OIPE
JAN 23 2004
PATENT & TRADEMARK OFFICE
JC85

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent application of:

Inventors: Lawrence Bain et al.
Serial No. 09/298,953
Filed: April 13, 1999
For: DYNAMIC-ADAPTIVE CLIENT-SIDE IMAGE MAP

RECEIVED
JAN 28 2004
Technology Center 2100

Examiner: Cong Lac T. Huynh
Group Art Unit: 2178
Docket No. 10990633-1
Date: January 16, 2004

BRIEF ON APPEAL

Commissioner of Patents and Trademarks
Washington, DC 20231

Sir:

This appeal is from the decision of the Examiner, in an Office Action mailed on July 15, 2003, finally rejecting claims 1-24.

REAL PARTY IN INTEREST

Hewlett-Packard Development Company, L.P. is the Assignee of the present patent application. Hewlett-Packard Development Company, L.P., is a Texas corporation with headquarters in Houston, Texas.

RELATED APPEALS AND INTERFERENCES

Applicants' representative has not identified, and does not know of, any other appeals of interferences which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

01/27/2004 MAHMED1 00000020 082025 09298953
01 FC:1402 330.00 DA

STATUS OF CLAIMS

Claims 1-24 are pending in the application. Claims 1-24 were finally rejected in the Office Action dated July 15, 2003. Applicants' appeal the final rejection of claims 1-24, which are copied in the attached Appendix I.

STATUS OF AMENDMENTS

An Amendment After Final is enclosed with this brief. Amendments have been made to the claims to correct a number of typographical errors and to place the claims into proper form for the Appeal.

SUMMARY OF INVENTION

The invention is described, below, with reference to Figures 1-5 of U.S. Application No. 09/298,953 ("Current Application") that are included in Appendix II. Figure 1 shows a display of an example web page, a description of which is transmitted through the Internet from a remote server computer to a user's client computer. The example web page display includes a "Read This First" text line 106, a "Quotes, News, Charts, Data" text line 108, and a number of additional charts and figures 102 and 110-113. Commonly, the information used by the user's client computer to display a web page is encoded within an HTML file transferred through the Internet to the client computer by a remote server computer. The HTML file may include references to image files and other information that is separately accessed by the client computer from the server computer and integrated with the display. For example, the displayed-chart image 110 may be encoded within a separate file, with indications of the identity of the separate file and the position at which the image encoded in the separate file is to be displayed within the web-page display encoded within the HTML file.

At the time that the Current Application was filed, as discussed in the Background of the Invention section and in the first few pages of the Detailed Description of the Invention section of the Current Application, several relatively recent innovations in web-page display were coming into general use, including active regions and dynamic images. Active regions are regions within the display of a web page that are associated with actions, such as displaying a different web page or image, or invoking a software routine. The active regions are specified by one of two types of image maps: (1) a client-side image map that is

generated and maintained on the user's client computer; and (2) a server-side image map that is maintained on a remote server. An image map can be thought of as the superposition of an abstract template over an image included on the web page, with the abstract template containing descriptions of the size, shape, and location of each active region as well as an association between the active region and an action. Figure 2 shows a larger scale representation of the left-hand, vertical column portion of the displayed web page shown in Figure 1, and Figure 3 shows a representation of an image map for the left-hand, vertical column portion of the displayed web page that includes specification of two active regions 302 and 304 corresponding to the "Read This First" text line (216 in Figure 2) and the "Quotes, News, Charts, Data" text line (218 in Figure 2). When a user employs a mouse or other input device to position a cursor displayed on the display device within an active region, the display of the cursor may change to indicate that further input of a mouse click, or other type of input, will invoke display of a different web page or image. For example, a mouse click input after positioning of a cursor over the "Read This First" text line (216 in Figure 2), specified by the image map active region 302, may result in display of a text file with additional user information overlying the initially displayed web page.

At the time that the Current Application was filed, locations within a displayed image were designated by device coordinates. For a computer display, imaginary horizontal and vertical axes incremented by pixels or by arbitrary horizontal and vertical unit increments are used to assign two-dimensional coordinates to each point in the display. In Figure 2, for example, the corners of the left-hand, vertical column portion of the displayed web page are labeled with device coordinates, such as device coordinates (0,0) that label the top, left-hand corner, and in Figure 3, the corners of the template and of the active regions 302 and 304 are also labeled with device coordinates. Device coordinates are also used by the operating system of a client computer to specify the location of a cursor moved by mouse input. Therefore, the position of a cursor can be correlated with a displayed image by comparing the device coordinates specifying the current cursor position with device coordinates specifying, among other things, active regions within the displayed image. For example, a routine running either on the client computer or on the remote server can determine that a mouse click is meant to activate display of a text file associated with active region 302 in Figure 3 by comparing the device coordinates corresponding to the current cursor position with the device coordinates specifying the corners of active region 302. Figure 4 shows the image map of Figure 3 superimposed over the left-hand, vertical column portion of the displayed

web page shown in Figure 2, as an illustration of the correspondence between the image map and the displayed web page which is obtained from the common use of device coordinates in both the image map and the displayed web page.

Server-side image maps involve a browser running on a client computer displaying a web page transmitting, using device coordinates, user input events that arise during display of the web page back to the server computer, followed by transmission of responses from the server computer to the client computer that facilitate any actions invoked by the user input. Client-side image maps, by contrast, are implemented, at run time, by web browsers, or by viewers invoked by web browsers, on client computers. Input to displayed images associated with client-side image maps thus does not incur the extra Internet traffic incurred in displaying images associated with server-side image maps.

Dynamically sizable and shiftable images within web pages are another example of relative recently developed enhanced graphic capabilities for web pages at the time that the Current Application was filed. A user viewing the display of a dynamic image within a web page may direct the browser to zoom into or out from the image to higher and lower resolutions, and to scroll the image in vertical and horizontal directions. The software mechanisms for describing and implementing dynamic images within web pages were separate and distinct from the mechanisms for describing and implementing client-side image maps.

At the time that the Current Application was filed, active-region functionality was not well integrated with dynamic image functionality. As a result, when client-side image maps were associated with dynamic images within web pages, the active regions were not automatically zoomed and shifted, or panned, along with the dynamic images. As a result, the active regions defined within the client-side image maps quickly lost their original correspondence to specific regions of the images with which they were associated. For example, as shown in Figure 5, when the scale of display of the left-hand, vertical column portion of the displayed web page 502 is enlarged, via a zoom operation, the image map 504 is not automatically correspondingly enlarged, leaving the specified active regions no longer overlying the portions of the displayed web page that they are intended to overlie. As a result, a mouse click input when the displayed cursor overlies the word "This" in the "Read This First" text line 510 would not result in the desired display of a text file, as it would prior to the zoom operation, as shown in Figure 4.

As discussed in the Summary of the Invention section of the Current Application, the present invention addresses the problem discussed in the previous paragraph by providing methods and systems for associating active regions with positions within displayed images in a device-and-display independent manner. In the Current Application, an embodiment of the present invention is described that represents an implementation of a dynamic-adaptive client-side image map that automatically tracks changes in a displayed dynamic image within a web page in order to maintain the originally specified correspondence between active regions defined by the dynamic-adaptive client-side image map and regions of the associated image displayed as part of the web page. In addition, the dynamic-adaptive client-side image map definitions are accessible to software routines invoked from web pages. In this embodiment of the present invention, an adaptive delivery module running on a server computer determines the capabilities of a client computer requesting a web page and, when the requesting computer's capabilities are compatible with inclusion of a dynamic-adaptive client-side image map, the adaptive delivery module transforms the HTML description of the requested web page to include one or more dynamic-adaptive client-side image maps and transmits the transformed HTML description of the web page to the requesting client computer. An enhanced web browser running on the requesting client computer receives the transformed web page and instantiates an appropriate image viewer based on tags within the transformed HTML description of the web page. The web browser provides the instantiated viewer with parameters describing any dynamic-adaptive client-side image maps within the transformed HTML description of the web page, and the instantiated viewer then processes the parameters to produce data structures that describe active regions within the image displayed by the viewer as part of display of the web page. The information contained within the data structures can be used at run time, during the display of the web page, by the instantiated viewer to correlate user input with active regions of the displayed image.

The Current Application provides a detailed, pseudo-code description of the client-side image viewer, encapsulated, in part, in the pseudo-code class "Image." As described in the Current Application beginning on line 34 of page 25:

> An Image object contains data members that represent the device coordinates of the upper left corner and lower left corner of the displayed images, "leftX," "leftY," "rightX," and "rightY" declared above on lines 4-7. In the current implementation, displayed images are assumed to be rectangular. These values are represented as floating point values to allow for precise multiplication by scale factors for zooming

operations. An Image object also contains a reference to the image, "img" declared above on line 8, and a reference to a MapInfo object, "map" declared above on line 9, that represents a DACSIM associated with the image. An instantiated Image object, in the current implementation, is an instantiated enhanced viewer.

One aspect of the described embodiment is the use, by the client-side image viewer invoked by a user's web browser on the user's client computer, of image-relative coordinates, as described beginning on line 5 of page 20 of the Current Application:

> The x,y image coordinates are stored as floating point values in data members "x" and "y" declared above on lines 4 and 5. Image coordinates range from 0 to 1. They are fractions of the width and height of an image, and are thus valid regardless of how the image is scaled or translated in device coordinates. These fraction coordinates are one embodiment of device-independent, image-relative coordinates, referred to as image coordinates.

As described in the Current Application, beginning on line 14 of page 19, the client-side image viewer employs instantiated objects of the class "Point" to represent defining points of active regions, and the class "Point," in turn, employs image-relative coordinates.

Thus, as described in the above-quoted passages from the Current Application, the client-side image viewer stores image dimensions in device coordinates, and stores the locations and positions in image-relative coordinates that are fractions of the image widths and heights containing the active regions. The image-relative coordinates therefore provide an automatic correspondence between active regions and displayed images. The client-side image viewer relates device coordinates to active regions using image-relative coordinates and by transforming the position of a mouse click, received from the operating system of the client computer in device coordinates, into image-relative coordinates prior to searching the active regions for an active region overlying the position of the mouse click, as described in the Current Application beginning on line 16 of page 25:

> The Image function "mouseClick" takes the device coordinate position of a mouse click input event, "x" and "y," as arguments and invokes an appropriate action if the mouse click was input to any active region within the image. First, on line 6, mouseClick determines whether the mouse click was input within the image. If so, then mouseClick sets the values of the local variables "imageX" and "imageY" to the image-relative coordinates of the mouse click, on lines 9 and 10, above. Then, in the *while* loop of lines 11-19, mouseClick determines whether the mouse click event was directed to any of the active regions via a call to the ImageArea member function "isInside" through an AreaInfo pointer stored in loop variable "a," on line 13, and, if so, invokes the mouse click action associated with the active region on line

15.

Applicants' described embodiments, as well as additional embodiments, are clearly claimed in the Current Application, as, for example, in claim 1:

> 1. A method for associating an **active region** with a corresponding position within an image included in a page displayed by a **browser running on a client computer**, the method comprising:
>
> sending a request by the browser to a server for a description of a page that includes a specification of the image and an associated client-side image map, the client-side image map specifying a shape, size, and location of the active region within the image and specifying actions to be performed in response to input events directed to the active region;
>
> receiving from the server in response to the request a description of the requested page that includes **an invocation of a viewer for displaying the image**, the invocation including parameters that describe the image and the client-side image map;
>
> **instantiating the viewer** and passing to the viewer the parameters included in the invocation;
>
> storing by the viewer **representations of active regions** within the image in **image-relative coordinates** along with indications of the actions to be performed in response to input events directed to the active region; and
>
> when an input event is detected by the browser during display of the page,
>
> **passing the input event by the browser to the viewer**, and
>
> when the viewer determines that the **input event was input to a position within the image corresponding to the active region**, determining an action specified for performance in response to the input event to the active region and calling for performance of the determined action. (emphasis added)

The aspects of Applicants' invention discussed in preceding paragraphs are clearly and straightforwardly claimed in claim 1, provided above. Applicants claim a method for associating an active region with a position within an image included in a page displayed by a client-side web browser. The client-side browser requests a page for display from a server, and receives, in response to the request, a description of the page that includes an invocation of a client-side image viewer for displaying the image and that includes parameters that are passed to the image-viewer, upon invocation, that describe the image to be displayed and a client-side image map used to correlate active regions within the image to the displayed image. Upon instantiation of the client-side image viewer on the client computer, the client-side viewer stores representations of active images within the image in image-relative coordinates, as discussed above. The client-side image viewer employs those representations to correlate input events passed from the browser to the image viewer with active regions, so that an input event directed to an active region within an image is properly handled.

ISSUES

1. Whether a reference that neither teaches, mentions, nor suggests including an invocation of a viewer in the description of a web page can serve as the basis for a 35 U.S.C. § 103(a) rejection of claims 1-24, all of which directly include, or include through dependency, specific language directed to inclusion of an invocation of a viewer in the description of a web page sent to a client computer for display.

2. Whether a reference that neither teaches, mentions, nor suggests instantiating a viewer by a client computer for display of an image included in the description of a web page can serve as the basis for a 35 U.S.C. § 103(a) rejection of claims 1-10, all of which directly include, or include through dependency, specific language directed to instantiating a viewer by a client computer for display of an image included in the description of a web page.

3. Whether a reference that neither teaches, mentions, nor suggests storing, by a viewer instantiated by a client-side browser, representations of active regions within an image for display as part of a displayed web page in image-relative coordinates can serve as the basis for a 35 U.S.C. § 103(a) rejection of claims 1-10 and 18-24, all of which directly include, or include through dependency, specific language directed to storing, by a viewer instantiated by a client-side browser, representations of active regions in image-relative coordinates.

4. Whether a reference that neither teaches, mentions, nor suggests passing an input event by a client-side browser to a client-side viewer and, when the viewer determines that the input event was input to a position within an image corresponding to the active region, determining by the viewer an action specified for performance in response to the input event to the active region and calling for performance of the determined action can serve as the basis for a 35 U.S.C. § 103(a) rejection of claims 1-10 and 18-24, all of which directly include, or include through dependency, specific language directed to passing an input event by a client-side browser to a client-side viewer and, when the viewer determines that the input event was input to a position within an image corresponding to an active region, determining by the viewer an action specified for performance in response to the input event to the active region and calling for performance of the determined action.

5. Whether a reference that neither teaches, mentions, nor suggests a server determining the capabilities for viewing pages of a browser and, when the browser is capable of accepting display altering commands from a user, substituting, in the description of the page, an invocation of a viewer for the specification of the image and the client-side image map included in the page to create a transformed page description for sending to the browser can serve as the basis for a 35 U.S.C. § 103(a) rejection of claims 11-17, all of which directly include, or include through dependency, specific language directed to a server determining the capabilities for viewing pages of a browser and, when the browser is capable of accepting display altering commands from a user, substituting, in the description of the page, an invocation of a viewer for the specification of the image and the client-side image map included in the page to create a transformed page description for sending to the browser.

6. Whether a reference that neither teaches, mentions, nor suggests data structures on a client computer that store image-relative indications of a particular portion of an image associated with an active region and actions to be performed in response to input events directed to the active region can serve as the basis for a 35 U.S.C. § 103(a) rejection of claims 18-24, all of which directly include, or include through dependency, specific language directed to data structures on a client computer that store image-relative indications of a particular portion of an image associated with an active region and actions to be performed in response to input events directed to the active region.

GROUPING OF CLAIMS

The claims fall into three groups, each group comprising a single independent claim and a number of dependent claims depending from the single independent claims: (I) claims 1-10; (II) claims 11-17; and (III) claims 18-24. The claims of groups I, II, and III stand or fall together with respect to issue (1). The claims of group I stand or fall together with respect to issues (2) and (4). The claims of groups I and III stand or fall together with respect to issue (3). The claims of group II stand or fall together with respect to issue (5). The claims of group III stand or fall together with respect to issue (6).

ARGUMENT

Claims 1-24 are currently pending in the application. In an Office Action dated July 15, 2003 ("Office Action"), the Examiner maintained a prior rejection of claims 1-24 under 35 U.S.C. § 103(a) as being unpatentable over Guedalia, U.S. Patent No. 6,356,283 B1 ("Guedalia"). Applicants respectfully disagree with these rejections.

All of the Examiner's rejections are based on Guedalia. Therefore, prior to presenting detailed arguments with respect to each of the above identified issues, Guedalia is first summarized, below:

Guedalia

Guedalia is discussed, in this and in following paragraphs, with reference to Figures 1-7 of Guedalia, provided in Appendix III. In the Abstract, Guedalia summarizes his disclosure:

> A method and system for archiving digital data on a server computer, and enabling a user, by means of a client computer, to interactively view a digital image derived from the digital data, the method including receiving an original HTML page by the client computer from the server computer, the original HTML page containing a view window within which a first image is displayed, the view window being partitioned into a plurality of sub-regions at least one of which contains a multiplicity of pixels, selecting a location within the view window corresponding to one of the plurality of sub-regions by the user, initial sending by the client computer to the server computer an indication of the sub-region selected by the user, creating by the server computer a new HTML page containing a link to an embedded image which corresponds to the indication, and further sending the new HTML page by the server computer to the client computer.

The above-described method and system is clearly reflected in the Figures. For example, Guedalia discloses a method and system involving server-side rendering and support for dynamic HTML pages, clearly shown in Figure 1 by the location of the software for dynamic HTML generation 16 located on the server computer 10. Figure 2 illustrates a server-side image map, in which, for example, active region, such as active region 48 bordered by dotted lines, is associated with a sub-region or sub-image 44. Guedalia describes image maps, beginning on line 35 of column 19, as follows:

> Internet browsers provide a mechanism, referred to as image maps, for carrying out this process. Image maps enable a browser to extract the coordinates of the location of the mouse pointer when the user clicks on the mouse, and send those coordinates back to the server. The server in turn receives these coordinates and calculates that sub-region within which they are situated. FIG. 2 illustrates a user clicking within sub-region 1.
>
> In order to calculate the sub-region within which the mouse coordinates are located, the server also needs to know the size and location of the view window; i.e. client state information. This client state information can also be sent from the client to the server, together with the mouse coordinates. Alternatively, the server can store this information in its

own memory.

Guedalia therefore prefers that mouse coordinates be processed into image-relative coordinates by the server, rather than the by the client-side browser.

In Figures 3 and 4, Guedalia shows flow charts that describe a preferred embodiment of his invention. Beginning on line 56 of column 19, Guedalia describes the client computer sending mouse-pointer coordinates to a server for interpretation of the mouse-pointer coordinates with respect to the displayed page containing subregions:

> Using a mouse or keyboard 34 for positioning, the user clicks on the mouse at a position within the view window. HTML identifies the image portion being viewed as an image map, and the client sends the mouse pointer coordinates 52 to the server computer 10. The server computer 10 calculates the sub-region within which the mouse coordinates are situated, and dynamically creates a new HTML page 54, with a link to an embedded image, constituting the response to the user mouse click.

Similarly, on line 31 of column 20, Guedalia describes the client computer sending mouse-pointer coordinates to a server for interpretation of the mouse-pointer coordinates with respect to the displayed page containing subregions:

> At step 72 the user viewing the page clicks on a location within the view window that contains the embedded image portion. The browser extracts the mouse pointer coordinates where the user clicked at step 74, and sends them to the server at step 76.
>
> At step 78 the server receives the mouse pointer coordinates, and at step 80 the server calculates which sub-region contains these coordinates.

An alternative embodiment is described with reference to Figures 5 and 6. In this alternative embodiment, the user's browser interprets mouse clicks, rather than the server computer, as described beginning on line 24 of column 21:

> At step 112 a user clicks on one of several buttons, each corresponding to a specific interactive command. Examples of such commands are (1) centered zoom-in, (2) centered zoom-out, (3) pan in one of several directions, (4) reset to initial image view and (5) print. At step 114 the browser identifies the area where the user clicked, thereby identifying the button and the desired associated user command. At step 116 the client sends the command data to the server in the form of an HTTP request.

However, Guedalia does not discuss or disclose how the user's browser interprets the mouse pointer coordinates or how the user identifies which subregions of the displayed page a mouse click is input to. When the client size browser interprets the mouse click and sends the command correlated with that interpretation, as, for example, for zooming out a particular image, in steps 50 and 110 in Figure 5, the server computer sends an entirely new HTML page, in step 54 of Figure 5, without including the magnified image, and the client computer

must then explicitly request the magnified image from the server, in step 58 of Figure 5, which the server returns in step 60 of Figure 5 for display by the client-side browser in step 62 of Figure 5. In Figure 6, these steps are numerically labeled 114, 116, 84, 96, 108, 94, respectively. *Therefore, Guedalia's client-side browser does not need to maintain a correspondence between an image map specifying active regions and displayed images. Guedalia's client-side browser receives an entirely new web page and new images displayed along with the web page after each command or operation invoked by a user's mouse input.*

Guedalia does use relative coordinates, defined by Guedalia beginning on line 14 of column 24. Guedalia mentions that portions of requested images are requested by a client-side browser from a server using relative coordinates. For example, Guedalia states, beginning on line 66 of column 23:

> Specifically, this IIP request is for data from a FlashPix image file photo.fpx using version 1.0 of the IIP protocol. The desired image data corresponds to a 300x300 pixel resolution image of the entire region of the FlashPix image, in the JPEG format. ... When the user clicks on this image, an HTTP request is dispatched to IP address 123.456.789.01 and port 8081. Port 8081 is the server port that handles incoming data. ... The server knows how to interpret this string as follows: ... The third data item, 0.0,0.0,1.0,1.0,300,300,0.0,0.0,1.0,1.0 is the current region of the FlashPix image being displayed. Regions are specified as relative coordinates x_1, y_1, x_2, y_2 where (x_1, y_1) are the coordinates of the lower left corner of the region ...

However, Guedalia does not teach, mention, or suggest storing image maps specifying active regions in terms of image-relative coordinates or transforming mouse pointer coordinates into image-relative coordinates. The client-side browser does not need to so. Instead, the client-side browser can use device-relative coordinates for all operations, since the client-side browser in Guedalia need not maintain a correspondence between active regions and displayed images. The browser in Guedalia need not maintain such a correspondence, because Guedalia receives a new HTML page description and new images for display following each operation or command solicited by mouse input.

Finally, Guedalia explicitly states, in lines 52-55 of column 4:

> A key feature of the present invention is that *it operates without the use of special client software, other than an Internet browser which is already resident on the client computer.* (emphasis added)

In other words, Guedalia explicitly states that a key feature of the technique disclosed in Guedalia is that no specialized client-side software is used, other than a standard Internet browser. Guedalia's browser does not, for example, invoke separate viewers or other image-

processing processes. Moreover, Guedalia's technique does not depend on specialized functionality embedded in the client-side browser. Instead, the server computer bears the burden of performing all image modifications needed for client-side, user-invoked, image-altering commands, such as zoom-in, zoom-out, and pan, by sending a new HTML page and new images to the client computer following each such operation initiated on the client compute by user input.

Arguments Directed to Each of the Six Issues

1. Whether a reference that neither teaches, mentions, nor suggests including an invocation of a viewer in the description of a web page can serve as the basis for a 35 U.S.C. § 103(a) rejection of claims 1-24, all of which directly include, or include through dependency, specific language directed to inclusion of an invocation of a viewer in the description of a web page sent to a client computer for display.

In claim 1 of the Current Application, Applicants claim a method in which a client-side browser receives "from the server in response to the request a description of the requested page that includes an invocation of a viewer for displaying the image, the invocation including parameters that describe the image and the client-side image map." An invocation included in the description of a web page directs the receiving browser to launch a separate viewer process that displays an image. This process is described in the Current Application beginning on line 20 of page 16:

> When the browser of the client computer receives a transformed HTML page from the server computer, the browser invokes an enhanced viewer, such as an ActiveX viewer, for each OpenPix image included in the transformed HTML description. The browser parses the HTML viewer invocation in order to extract the various parameters included within the invocation and passes those parameters to the enhanced viewer during instantiation of the viewer. The browser then displays the web page described by the received transformed HTML description and the instantiated enhanced viewers display the OpenPix images included within the HTML description.

A specific example of inclusion of an image-viewer invocation within an HTML web-page description is provided in the Current Application, beginning on line 24 of page 13:

> The HTML extract, provided below, represents the HTML code substituted by the ADM for the HTML descriptions of the image shown in Figures 2, 4 and 5, provided

above, in the case that the server is configured to deliver an ActiveX™ viewer to the browser of the requesting client computer and the browser of the requesting client computer is capable of using the ActiveX viewer to display an OpenPix image:

```
 1 <object ID="Fviewer" WIDTH="116" HEIGHT="450"
 2 CLASSID="CLSID:56ECEF01-E59E-11DO-9243-000000000000"
 3 CODEBASE="/OpenPix/controls/ViewPix01.CAB#Version=0,2,3,6">
 4 <param name="SourceURL"
 5 value="/opx-bix/OpxIIPISA.d11?FIF=/art/quotesreadthis.jpg">
 6 <param name="MouseDefaults" value="N">
 7 <PARAM NAME="OPX_CLIMAP_AREAS"
 8 VALUE="OPX_VIEWER2^^myMap^^
 9 Area_01^^rectangle^^10,2,104,15^^/help/disclaimer.htm.htm ~^^]]^^
10 Area_02^^rectangle^^8,23,107,42^^http://quote.fool.com~^^]]^^#>
11 </object>
```

In the above HTML enhanced viewer, the enhanced viewer invocation is introduced by the symbol "<object" on line 1 and terminated by the symbol "/object>" on line 11. This object tag includes, on line 1, attributes that identify the enhanced viewer that will display the image as well as the width and height of the displayed image in device coordinates.

Finally, beginning on line 1 on page 17, Applicants reference Figure 7 of the Current Application in which Applicants illustrate an implementation of a viewer:

> Figure 7 is a flow control diagram of the operation of an enhanced viewer as instantiated by a browser for displaying OpenPix images with an associated DACSIM. In step 702, the enhanced viewer is instantiated and, as part of the instantiation process, receives a list of parameters that specify the image to be displayed, the initial width, height, and location of the image on the display device of the client computer on which the enhanced viewer is running, and parameters that define a DACSIM associated with the image. The enhanced viewer instantiates data structures to contain information stored in the received parameters.

Invocation of a viewer by a client-side browser therefore involves launching a separate viewer process, on the client computer, for displaying an image. The server specifies to the browser which viewer to invoke, as well as a number of parameters that indicate which image to display, and how to initially display it.

By contrast, Guedalia explicitly states, as discussed above, that "[a] key feature of the present invention is that it operates without the use of special client software, other than an Internet browser which is already resident on the client computer." Guedalia therefore explicitly teaches away from Applicants' approach of invoking a separate viewer on the client-side for displaying images. Guedalia does not employ, and does not need, a separate viewer because all image processing is done by the server computer in Guedalia's system, as discussed above. For example, beginning on line 36 of column 4, Guedalia describes his approach to interactive display of images within a web-page display:

> The present invention also enables a client to interact with an image without the use of image maps, by selecting one of several commands, such as "zoom-in," "zoom-out," "pan," and "reset." The command selected is embedded within an HTTP request dispatched from the client to the server, which the server can parse to extract the selected command. The server uses this command to modify the HTML page, in order to update the image reference contained therein in accordance with the selected command.

Guedalia includes a number of HTML extracts, not one of which includes a viewer invocation. Guedalia makes no mention of a client-side viewer, and no mention of additional software processes on the client side other than a browser. Guedalia in no way teaches, mentions, or suggests Applicants' claimed inclusion of an image-viewer invocation within a description of a web page sent from a server computer to a client-side browser.

2. Whether a reference that neither teaches, mentions, nor suggests instantiating a viewer by a client computer for display of an image included in the description of a web page can serve as the basis for a 35 U.S.C. § 103(a) rejection of claims 1-10, all of which directly include, or include through dependency, specific language directed to instantiating a viewer by a client computer for display of an image included in the description of a web page.

In claim 1 of the Current Application, Applicants claim a method in which a client-side browser instantiates a viewer and passes to the viewer parameters included in an invocation of the viewer received in the description of a web page, and in which the viewer, once instantiated, stores representations of active regions within the image in image-relative coordinates along with indications of the actions to be performed in response to input events directed to the active region. When an input event is detected by the browser during display of the page, the browser passes the input event to the viewer and, when the viewer determines that the input event was input to a position within the image corresponding to an active region, the viewer determines an action specified for performance in response to the input event to the active region and calling for performance of the determined action. In the above discussion of issue(1), portions of the Current Application describing invocation of the viewer are quoted. Invocation of a viewer by a client-side browser involves launching a separate viewer process, on the client computer, for displaying an image. The server specifies to the browser which viewer to invoke, as well as a number of parameters that indicate which image to display, and how to initially display it. Moreover, as clearly claimed in claim 1, the client-side browser passes input events to the viewer, so that the viewer can

respond to input events input to active regions.

By contrast, as discussed above, in Guedalia's disclosed system, Guedalia's client-side browser passes input events not to an invoked image viewer, but back to the remote server, which then furnishes new images and web page descriptions for display by the client-side browser. This aspect of Guedalia's system is described in the quoted portion of Guedalia's Summary in the above discussion of issue(1), and is clearly and unambiguously shown in Figures 3-6, and discussed in Guedalia with reference to those Figures.

In response to Applicants' representative's arguments in several Responses, the Examiner concludes, in the Advisory Action dates October 20, 2003 ("Advisory Action"):

> As admitted by Applicants, Guedalia discloses the server-side techniques. Guedalia also discloses zooming and spanning operations operated on the client side (col 22, line 62 to col 23, line 53). Therefore, Guedalia is involved with both client-side and server-side implementations as argued. Also, according to the Applicants, the client web browser of the current application is enhanced and is different from the standard web browser since said enhanced web browser includes the zooming and scrolling operations (Remarks, pages 4-5). Guedalia has zooming and spanning features that can be applied to the images to alter the images at the client (col 4, lines 36-51; col 16 lines 35-51). The web browser in Guedalia, therefore, is an enhanced web browser as argued. ... Though Guedalia explicitly states that Guedalia's "present invention provides a novel approach to clientless HTML-driven interactive image navigation over the Internet", Guedalia further states that "the present invention also enables a client top interact with an image ... by selecting one of several commands, such as "zoom-in", "zoom-out", "pan" ... a "pan" command can be used to translate the image being displayed by a fixed pre-determined amount ... "(col 4, lines 36-51). That means, Guedalia is also concerned with the client-side functionality providing the capability of interacting with images at client.

Applicants respectfully observe that the Examiner has failed to fully consider the claim language, the Current Application, and the arguments presented in the Responses. Guedalia indeed provides the ability for a user to zoom and pan images displayed as part of a displayed web page on a client computer. However, as discussed above, Guedalia accomplishes this dynamic image display primarily on the server, with the server furnishing new images for display following receiving, from the client-side browser, each user-initiated zoom or pan operation as direct user input or as image-display commands as interpreted from user input by the client-side browser. By contrast, in Applicants' claimed method, the client-side browser passes input events not to the server, but to an image viewer invoked by the client-side browser for display of the image. In Applicants' claimed method, image display operations are carried out on the client computer, rather than on the server. It is Guedalia's expressed intent to carry out image operations on the server computer, well summarized in Guedalia's

statement that "[a] key feature of the present invention is that it operates without the use of special client software, other than an Internet browser which is already resident on the client computer."

In short, Guedalia's neither teaches, mentions, or suggests client-side image display by an image-viewer invoked by a client-side browser, but an oppositely oriented system in which image operations are performed on the server, and furnished to a standard web browser, so that dynamic images can be obtained on a client computer without an enhanced web browser and without invocation of additional software processes. Although the Examiner appears to argue that, since similar results are obtained, Applicants' claimed method and system are obvious. In Applicants' representative's opinion, that reasoning is flawed. The Examiner must show a teaching or suggestion in Guedalia that makes Applicants' claimed method and system obvious. Showing that obtained results are similar is not sufficient. The fact that Guedalia teaches away from Applicants' claimed method and system is extremely significant. Furthermore, the results of the two methods are not as similar as the Examiner would suggest. In Guedalia's system, each user input, such as a mouse click, to an active region incurs two round-trip interactions between the client computer and the server computer, leading to very high data transmission rates over the communications links interconnecting them, and leading to significant lag times between a user input and a displayed response to that input. By contrast, in Applicants' claimed system, the image-display operations are carried out entirely on the client computer, avoiding the bandwidth overhead and lag-times associated with client-computer/server-computer intercommunication.

3. Whether a reference that neither teaches, mentions, nor suggests storing, by a viewer instantiated by a client-side browser, representations of active regions within an image for display as part of a displayed web page in image-relative coordinates can serve as the basis for a 35 U.S.C. § 103(a) rejection of claims 1-10 and 18-24, all of which directly include, or include through dependency, specific language directed to storing, by a viewer instantiated by a client-side browser, representations of active regions in image-relative coordinates.

In claim 1 of the Current Application, Applicants claim a method in which a client-side image viewer stores representations of active regions within the image in image-relative coordinates, and, in claim 18, Applicants claim a system in which data structures on the client

computer store image-relative indications of the particular portion of the image associated with the active region and actions and actions to be performed in response to input events directed to the active region. Image-relative coordinates are described beginning on line 5 of page 20 of the Current Application, as discussed above. An exemplary data structure used for storing defining points for active regions in image-relative coordinates is described, in the Current Application, beginning on line 14 of page 19:

> First, an enumeration defining the various active region shapes and a class that defines a Point object that stores, in image coordinates, a defining point of an active region are provided below:
> enum areaType = {DEFAULT, CIRCLE, POLY, RECT};
>
> ```
> 1 class Point
> 2 {
> 3       protected:
> 4               double x;
> 5               double y;
> 6               Point* next;
> 7
> 8       public:
> 9               double getX() const;
> 10              void    setX (const double x);
> 11              double getY() const;
> 12              void    setY (const double y);
> 13              Point*  getNext() const;
> 14              void    setNext (const Point* nxt);
> 15              Point (const double x, const double y, const Point* nxt);
> 16              Point (const double x, const double y);
> 17              ~Point();
> 18 };
> ```
>
> Instantiated Point objects appear as objects 814-820 in Figure 8. The x,y image coordinates are stored as floating point values in data members "x" and "y" declared above on lines 4 and 5. Image coordinates range from 0 to 1. They are fractions of the width and height of an image, and are thus valid regardless of how the image is scaled or translated in device coordinates. These fraction coordinates are one embodiment of device-independent, image-relative coordinates, referred to as image coordinates.

By contrast, as discussed above, Guedalia neither teaches, mentions, nor suggests storing of image-relative coordinates in a data structure, in association with actions to be performed upon user input to the active region. In fact, the only mention of image-relative coordinates in Guedalia is use of image-relative coordinates in a request sent from a client-browser to a server for return of magnified sub-images, where the sub-images are specified in terms of image-relative coordinates, as discussed above. Because Guedalia does not perform image operations on the client computer, as discussed above, there is no need for Guedalia's

client-side browser to maintain a correspondence between displayed images and active regions. Instead, as discussed above, Guedalia's client-side browser receives a new web-page description and new images for display from the server following each user input that affects display of images.

In response to Applicants' representative's arguments in several Responses, the Examiner concludes, in the Advisory Action dates October 20, 2003 ("Advisory Action"):

> Examiner respectfully disagrees.
> Guedalia discloses:
> - image-relative coordinates on the client computer (col 23, line 54 to col 24, line 48)
> - storing them on the client computer (col 4, lines 9-35: "... thus by caching these images within the client, the client provides an instant interactive response whenever the user navigates back to the same image..")
> - client-side functionality employing image-relative coordinates (col 4, lines 36-51; col 16, lines 35-60)

As discussed above, the cited text (col 23, line 54 to col 24, line 48) discusses using image-relative coordinates to specify sub-images in an HTTP request sent from the client computer to the server computer. This has nothing to do with storing definitions of active regions on the client computer for use in correlating user input with displayed images. Including image-relative coordinates in an HTTP request does not constitute a client-side image viewer that stores representations of active regions within the image in image-relative coordinates, as clearly claimed in claim 1. Of course, as discussed above, there is also no image-viewer in Guedalia to do the storing. There is not a single mention or suggestion of image-relative coordinates in the remaining cited portions of Guedalia (col 4, lines 9-35) and (col 4, lines 36-51; col 16, lines 35-60). Caching images has nothing to do with storing definitions of active regions in image-relative coordinates. Images are routinely, if not always, stored in many applications without the use of image-relative coordinates. The Examiner seems to assume that, because images are stored in Guedalia's system, and since various image-display-altering operations are performed by Guedalia's server computer, image-relative coordinates must necessarily be used. But, as discussed above, that is not at all the case. It is quite common to perform such operations using device-relative coordinates. There is nothing in Guedalia to suggest storing active-region definitions in image-relative coordinates, and there is no suggestion for a need to do so.

4. Whether a reference that neither teaches, mentions, nor suggests passing an input event by a client-side browser to a client-side viewer and, when the viewer determines that the input event was input to a position within an image corresponding to the active region, determining by the viewer an action specified for performance in response to the input event to the active region and calling for performance of the determined action can serve as the basis for a 35 U.S.C. § 103(a) rejection of claims 1-10 and 18-24, all of which directly include, or include through dependency, specific language directed to passing an input event by a client-side browser to a client-side viewer and, when the viewer determines that the input event was input to a position within an image corresponding to an active region, determining by the viewer an action specified for performance in response to the input event to the active region and calling for performance of the determined action.

In claim 1 of the Current Application, Applicants claim a method in which, when an input event is detected by the browser during display of the page, the client-side browser passes the input event to a client-side image viewer invoked by the browser, and, when the viewer determines that the input event was input to a position within the image corresponding to the active region, the image viewer determines an action specified for performance in response to the input event to the active region and calls for performance of the determined action. As discussed above in great detail, in Applicants' system, a client-side browser invokes a separate client-side image viewer to display dynamic images within a web page, and the client-side browser, after instantiating the image viewer, passes user input to the image viewer for correlation with active regions and for appropriate handling. As discussed above, in detail, Guedalia's system does not use an image-viewer, or any software process other than a standard web browser, on the client side. User input is passed, in Guedalia's system, to the remote server through the Internet, and the remote server carries out the image-display-altering operations, sending the resulting images and web-page descriptions back to the client computer through the Internet. Guedalia not only fails to teach this clearly claimed aspect of Applicants' method, but actually explicitly teaches away from this clearly claimed aspect of Applicants' method.

5. Whether a reference that neither teaches, mentions, nor suggests a server determining the capabilities for viewing pages of a browser and, when the browser is capable of accepting display altering commands from a user, substituting, in the description of the page, an

invocation of a viewer for the specification of the image and the client-side image map included in the page to create a transformed page description for sending to the browser can serve as the basis for a 35 U.S.C. § 103(a) rejection of claims 11-17, all of which directly include, or include through dependency, specific language directed to a server determining the capabilities for viewing pages of a browser and, when the browser is capable of accepting display altering commands from a user, substituting, in the description of the page, an invocation of a viewer for the specification of the image and the client-side image map included in the page to create a transformed page description for sending to the browser.

In claim 11 of the Current Application, Applicants claim a method in which a server determines the capabilities for viewing pages provided by a browser running on a client computer and, when the browser, running on the client computer, is capable of accepting display altering commands from a user while displaying a page, parses the description of the page to find the specification of the image and the client-side image map included in the page, substitutes, in the description of the page, an invocation of a viewer for the specification of the image and the client-side image map included in the page, including in the invocation parameters that specify the image and the client-side image map, to create a transformed page description, and sends the transformed page description to the browser. These steps are clearly shown in Figure 6 of the Current Application as steps 606, 608, and 610. In discussion of issue (1), above, sections of the Current Application showing how the invocation is included into the description of a web page are quoted.

By contrast, Guedalia neither teaches, suggest, nor mentions a server determining the image-viewing capabilities of a client browser, substituting an image viewer invocation for an image specification in the description of a web page, or sending a transformed web page to a client-side browser. As discussed above, this lack of teaching, mention, or suggestion is quite understandable in view of the fact that Guedalia's system does not use a client-side image viewer, but instead employs only a standard web browser on the client computer, based on Guedalia's explicitly stated intent to employ only a standard web browser on the client computer, and nothing more. The fact that, in Applicants' method, an image-viewer invocation is substituted for an image specification is indicative of the fact that an image specification is not the same as an image-viewer invocation. Neither Figure 4 nor Figure 6 of Guedalia show steps equivalent to, or even merely suggestive of, steps 606, 608, and 610 in Figure 6 of the Current Application. Were Applicants' server to know, beforehand, that only

standard web browsers are available on the client computer, steps 606, 608, and 610 would be unnecessary, since only a standard image display capability would be present on the client computer. None of the HTML extracts shown in Guedalia contain an invocation of an image viewer. Guedalia is wholly and completely unrelated to this claimed aspect of the Applicants' method and system.

.

6. Whether a reference that neither teaches, mentions, nor suggests data structures on a client computer that store image-relative indications of a particular portion of an image associated with an active region and actions to be performed in response to input events directed to the active region can serve as the basis for a 35 U.S.C. § 103(a) rejection of claims 18-24, all of which directly include, or include through dependency, specific language directed to data structures on a client computer that store image-relative indications of a particular portion of an image associated with an active region and actions to be performed in response to input events directed to the active region.

In claim 18 of the Current Application, Applicants clearly claim a system in which data structures on a client computer that store image-relative indications of the particular portion of the image associated with the active region and actions to be performed in response to input events directed to the active region. This issue is similar, although distinct, from issue (3), discussed above. In the claimed system, image-relative coordinates are stored for active regions, in association with specification of actions related to the active regions, in data structures maintained by an image viewer, in order maintain a correspondence between displayed images, which may be altered dynamically, and active regions. In the discussion of issue (3), an example implementation of an image-relative-coordinate-storing data structure is provided, and the pseudocode implementation included in the Current Application provides a detailed implementation of complete, exemplary data structures. As discussed above, Guedalia neither teaches, mentions, nor suggests storing image-relative coordinates on the client computer, nor any reason for doing so.

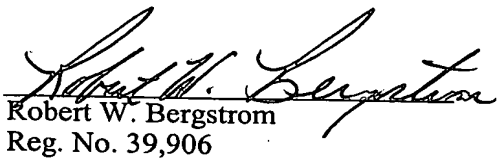
CONCLUSION

The Examiner has cited a single reference, Guedalia, in support of a rejection of claims 1-24 under 35 U.S.C. § 103(a). The Examiner needs to show that Guedalia would

have suggested, to one of ordinary skill in the art that Applicants' claimed methods and system should be carried out and implemented, respectively, and that Applicants' claimed methods and system would have a reasonable likelihood for success if practiced and implemented, respectively. However, Guedalia explicitly teaches away from Applicants' claimed methods and system, employing an approach to providing dynamic display of images within web pages entirely different from Applicants' claim methods and system. Guedalia explicitly states that intention, and explicitly states advantages of that approach, most significant of which is the ability to provide dynamic images by using only a standard browser on a client computer. Guedalia fails to teach, mention, or suggest the majority of the elements in Applicants' claims, as discussed above. Where Guedalia does not teach away from Applicants' claimed methods and system, Guedalia is generally unrelated to Applicants' claimed methods and system. In Applicants' view, Guedalia cannot be cited in support of 35 U.S.C. § 103(a) rejections of Applicants' claimed methods and system, and the 35 U.S.C. § 103(a) rejection of claims 1-24 are therefore improper.

Applicants' respectfully submit that all statutory requirements are met and that the present application is allowable over all the references of record. Therefore, Applicants' respectfully requests that the present application be passed to issue.

Respectfully submitted,
Lawrence Bain et al.
OLYMPIC PATENT WORKS PLLC

By 
Robert W. Bergstrom
Reg. No. 39,906

Olympic Patent Works PLLC
P.O. Box 4277
Seattle, WA 98104
206.621.1933 telephone
206.621.5302 fax

APPENDIX I

1. A method for associating an active region with a corresponding position within an image included in a page displayed by a browser running on a client computer, the method comprising:

sending a request by the browser to a server for a description of a page that includes a specification of the image and an associated client-side image map, the client-side image map specifying a shape, size, and location of the active region within the image and specifying actions to be performed in response to input events directed to the active region;

receiving from the server in response to the request a description of the requested page that includes an invocation of a viewer for displaying the image, the invocation including parameters that describe the image and the client-side image map;

instantiating the viewer and passing to the viewer the parameters included in the invocation;

storing by the viewer representations of active regions within the image in image-relative coordinates along with indications of the actions to be performed in response to input events directed to the active region; and

when an input event is detected by the browser during display of the page,

passing the input event by the browser to the viewer, and

when the viewer determines that the input event was input to a position within the image corresponding to the active region, determining an action specified for performance in response to the input event to the active region and calling for performance of the determined action.

2. The method of claim 1 wherein the page displayed by the browser running on a client computer is a web page.

3. The method of claim 2 wherein the server runs on a server computer and a description of the web page is requested by the browser from the server and received by the browser from the server via the Internet.

4. The method of claim 2 wherein the server runs on the client computer and a description of the web page is requested by the browser from the server and received by the

browser from the server via an inter-process communications medium within the client computer.

5. The method of claim 2 wherein the description of the web page received from the server in response to the request by the browser is a hyper-text markup language document.

6. The method of claim 2 wherein the image is an OpenPix image and wherein an invocation to a browser extension image viewer is included in the description of the web page.

7. The method of claim 2 wherein input events directed to the active region may include mouse-click, mouse-into, and mouse-out-from events, and actions to be performed in response to input events include display of a web page, display of an image, or launching of a software routine.

8. The method of claim 2 where image-relative coordinates represent the position of points within the image, a point within the image represented by a pair of coordinates, a first coordinate of the pair having a fractional value representing the ratio of a horizontal line segment to a horizontal dimension of the image with a first endpoint coincident with a vertical edge of the image and a second endpoint coincident with the point, the horizontal line segment perpendicular to the vertical edge of the image, the second coordinate of the pair having a fractional value representing the ratio of a vertical line segment to a vertical dimension of the image with a first endpoint coincident with a horizontal edge of the image and a second endpoint coincident with the point, the vertical line segment perpendicular to the horizontal edge of the image, the horizontal and vertical edges of the image intersecting at an origin having coordinates (0, 0).

9. The method of claim 2 further including:

when a display altering input event is detected by the browser,

passing a display altering input command by the browser to the viewer, and

altering the display of the image by the viewer in accordance with the input command.

10. The method of claim 9 wherein display altering input events include a zoom input event and a pan input event.

11. A method for serving a description of a page from a server to a browser running on a client computer that requests the page, the description of the page provided to the browser by the server containing an invocation of a viewer, the invocation including parameters that specify an image included in the page and an active region within the image, the method comprising:

receiving a request from the browser by the server for a description of the page that includes a specification of the image and an associated client-side image map, the client-side image map specifying a shape, size, and location of the active region within the image and that specifies actions to be performed in response to input events directed to the active region;

retrieving a description of the page;

determining the capabilities for viewing pages provided by the browser running on the client computer; and

when the browser, running on the client computer, is capable of accepting display altering commands from a user while displaying a page,

parsing the description of the page to find the specification of the image and the client-side image map included in the page,

substituting, in the description of the page, an invocation of a viewer for the specification of the image and the client-side image map included in the page, including in the invocation parameters that specify the image and the client-side image map, to create a transformed page description, and

sending the transformed page description to the browser.

12. The method of claim 11 wherein the page requested from the server by the browser running on a client computer is a web page.

13. The method of claim 12 wherein the server runs on a server computer and a description of the web page is requested by the browser from the server and sent by the server to the browser via the Internet.

14. The method of claim 12 wherein the server runs on the client computer and a description of the web page is requested by the browser from the server and sent by the server to the browser via an inter-process communications medium within the client computer.

15. The method of claim 12 wherein the description of the web page retrieved by the server in response to the request by the browser is a hyper-text markup language document.

16. The method of claim 12 wherein the image is an OpenPix image and wherein an invocation to a browser extension image viewer is included in the description of the web page.

17. The method of claim 12 wherein input events directed to the active region may include mouse-click, mouse-into, and mouse-out-from events, and actions to be performed in response to input events include display of a web page, display of an image, or launching of a software routine.

18. A system for displaying a page that includes an image and an active region correlated with a particular portion of the image, the display of the page modifiable during the display of the page on a display device of a client computer such that the active region within the image remains correlated with the portion of the image, the system comprising:

a browser running on the client computer that displays the page;

a server that receives a request from the browser for a description of the page and that that provides a description of the page that contains an invocation of a viewer, the invocation including parameters that specify an image included in the page and an active region within the image; and

data structures on the client computer that store image-relative indications of the particular portion of the image associated with the active region and actions and actions to be performed in response to input events directed to the active region.

19. The method of claim 19 wherein the page displayed by the browser running on a client computer is a web page.

20. The method of claim 19 wherein the server runs on a server computer and a description of the web page is requested by the browser from the server and received by the browser from the server via the Internet.

21. The method of claim 19 wherein the server runs on the client computer and a description of the web page is requested by the browser from the server and received by the browser from the server via an inter-process communications medium within the client computer.

22. The method of claim 19 wherein the description of the web page received from the server in response to the request by the browser is a hyper-text markup language document.

23. The method of claim 19 wherein input events directed to the active region may include mouse-click, mouse-into, and mouse-out-from events, and actions to be performed in response to input events include display of a web page, display of an image, or launching of a software routine.

24. The method of claim 19 where image-relative coordinates represent the position of points within the image, a point within the image represented by a pair of coordinates, a first coordinate of the pair having a fractional value representing the ratio of a horizontal line segment to a horizontal dimension of the image with a first endpoint coincident with a vertical edge of the image and a second endpoint coincident with the point, the horizontal line segment perpendicular to the vertical edge of the image, the second coordinate of the pair having a fractional value representing the ratio of a vertical line segment to a vertical dimension of the image with a first endpoint coincident with a horizontal edge of the image and a second endpoint coincident with the point, the vertical line segment perpendicular to

the horizontal edge of the image, the horizontal and vertical edges of the image intersecting at an origin having coordinates (0, 0).

APPENDIX II

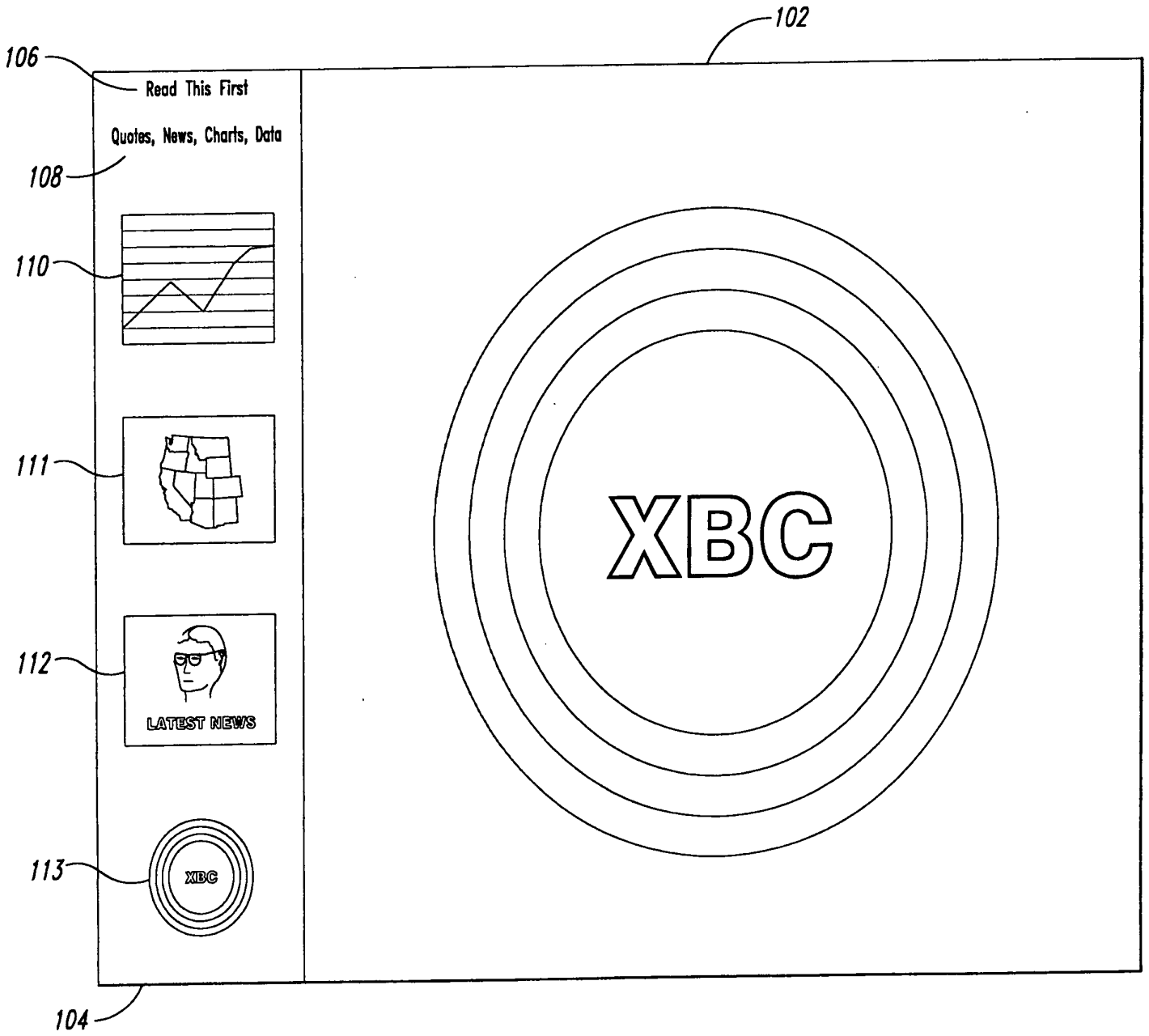

102
106
Read This First
Quotes, News, Charts, Data
108
110
111
112
LATEST NEWS
113
XBC
XBC
104

Fig. 1

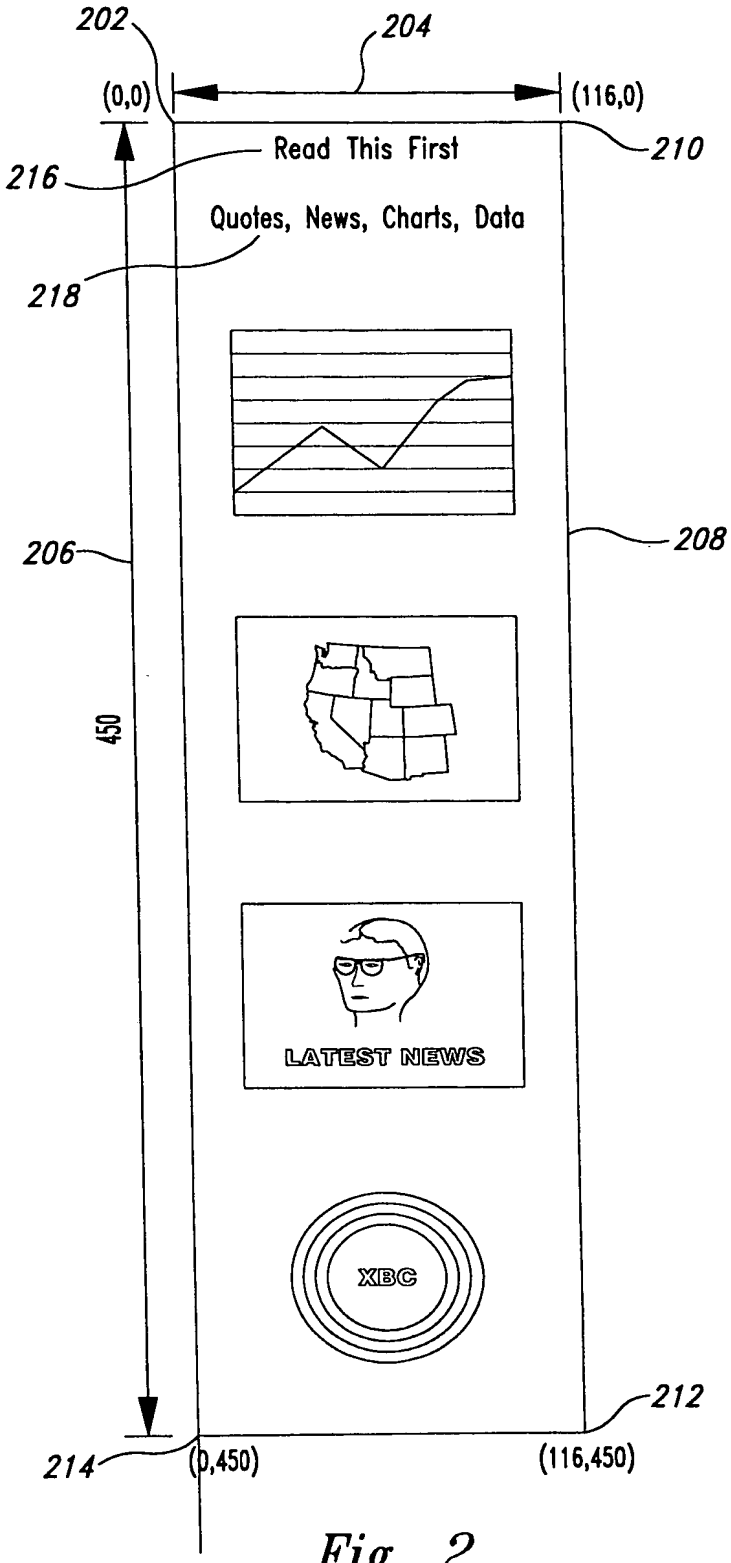

202
204
(0,0)
(116,0)
Read This First
210
216
Quotes, News, Charts, Data
218
206
208
450
LATEST NEWS
XBC
212
214
(0,450)
(116,450)

Fig. 2

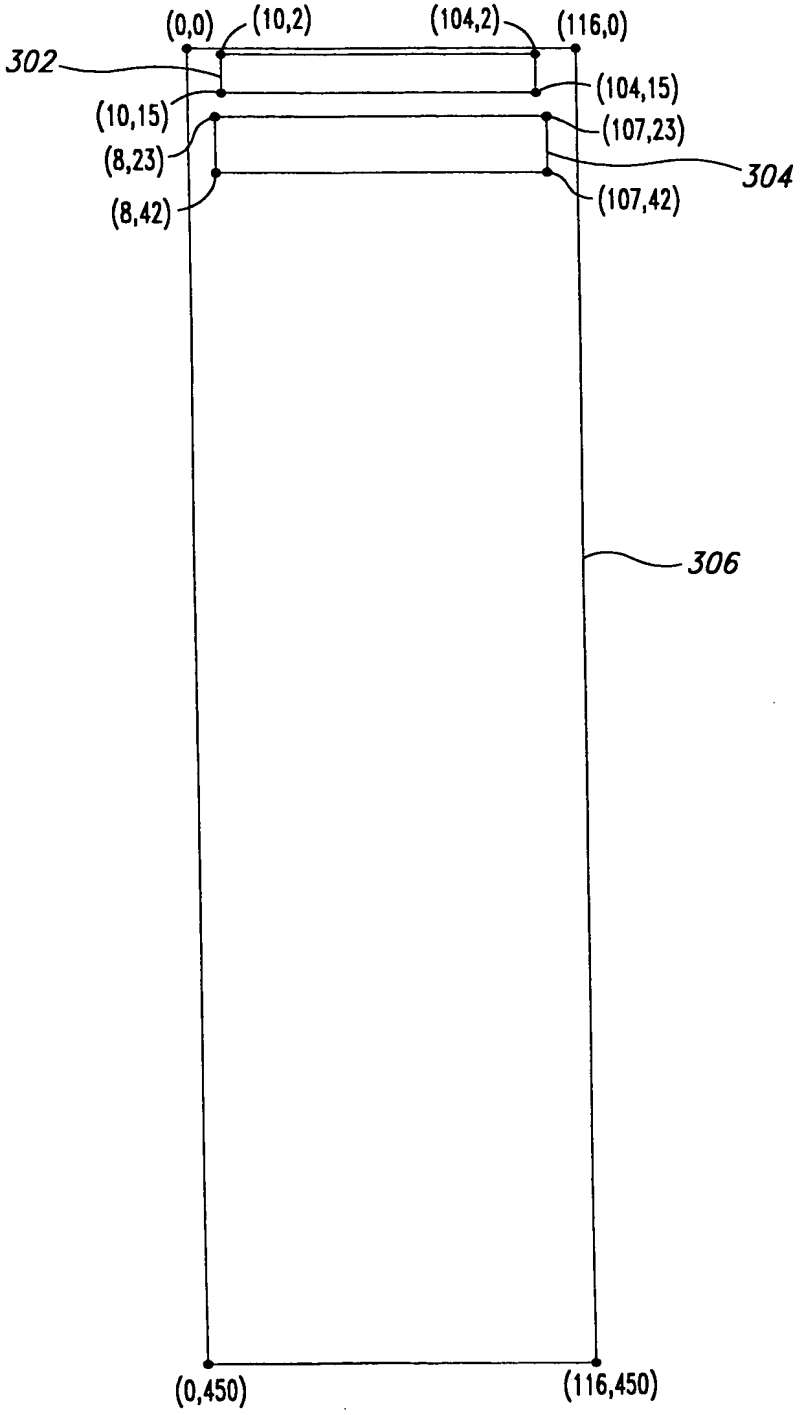

(0,0)
(10,2)
(104,2)
(116,0)
302
(104,15)
(10,15)
(107,23)
(8,23)
304
(8,42)
(107,42)
306
(0,450)
(116,450)

Fig. 3

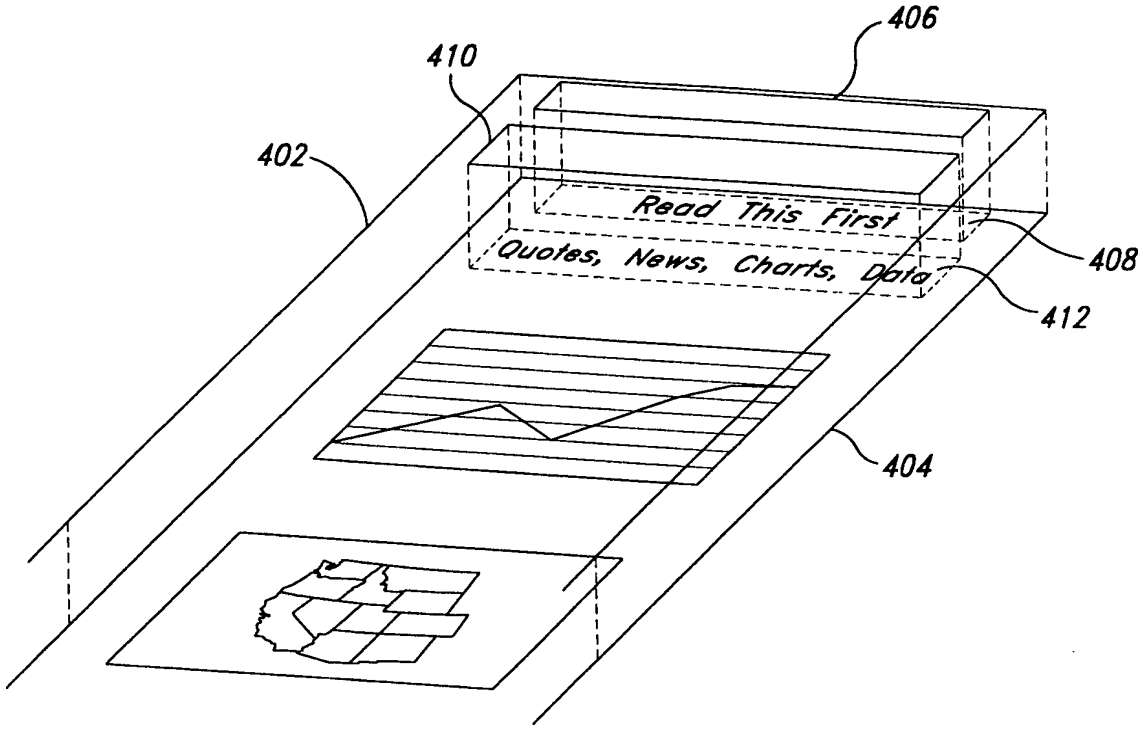

406
410
402
Read This First
Quotes, News, Charts, Data
408
412
404

Fig. 4

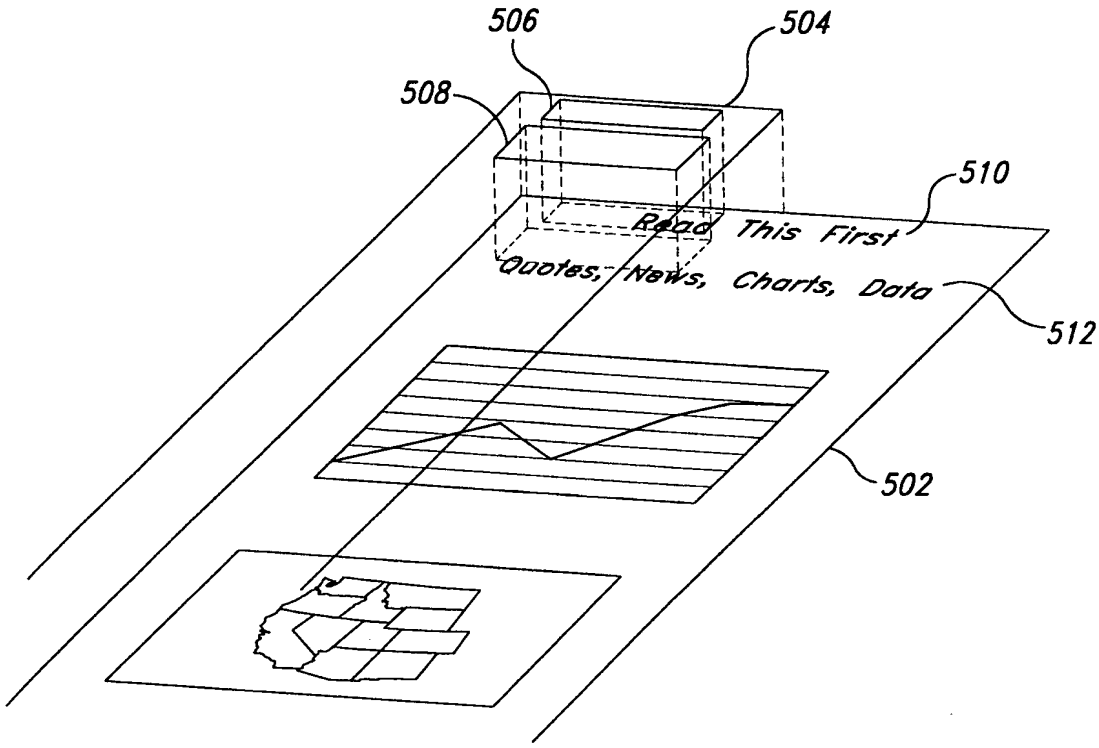

506
504
508
510
Read This First
Quotes, News, Charts, Data
512
502

Fig. 5

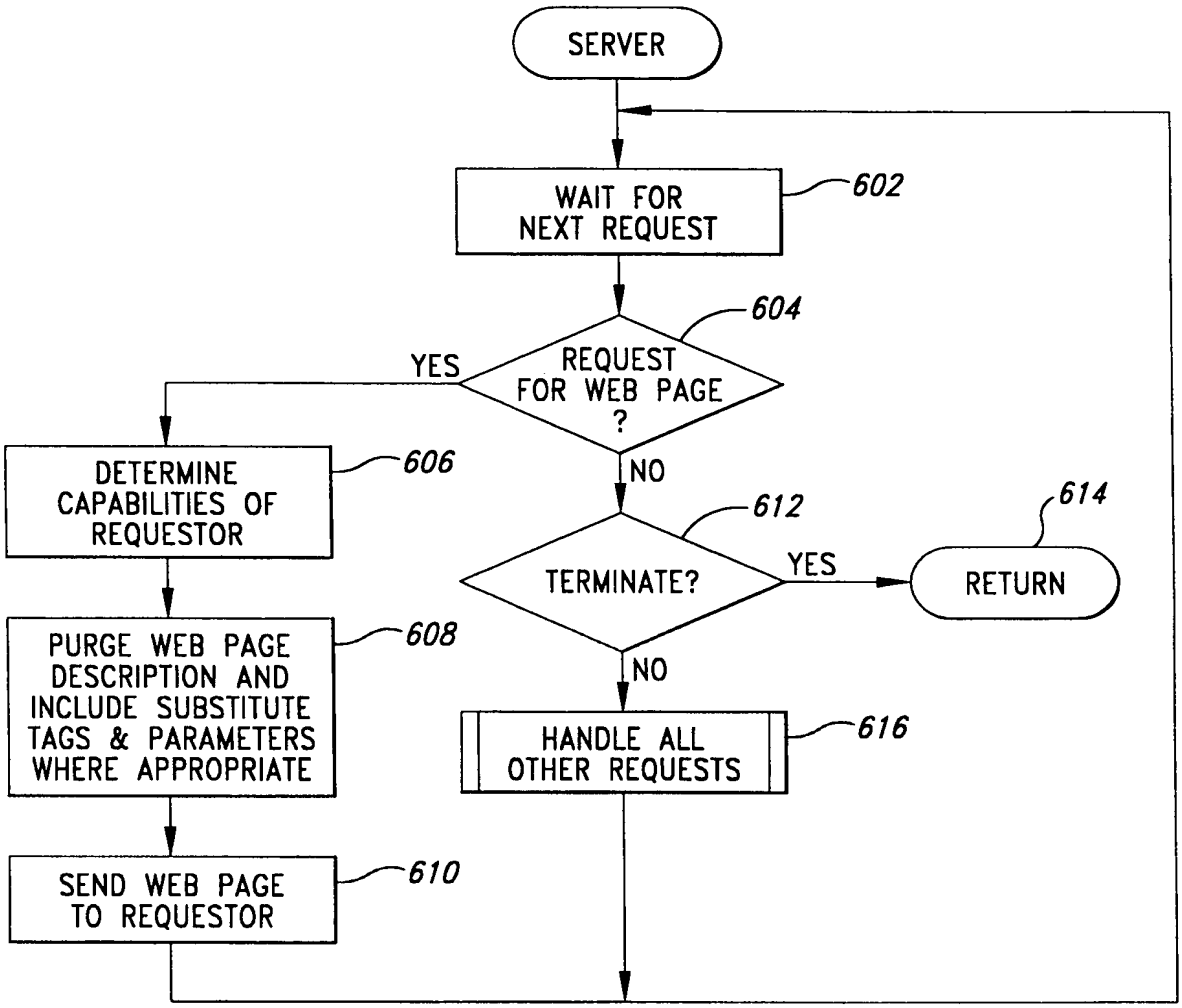

SERVER
WAIT FOR NEXT REQUEST
602
REQUEST FOR WEB PAGE ?
604
YES
NO
DETERMINE CAPABILITIES OF REQUESTOR
606
PURGE WEB PAGE DESCRIPTION AND INCLUDE SUBSTITUTE TAGS & PARAMETERS WHERE APPROPRIATE
608
SEND WEB PAGE TO REQUESTOR
610
TERMINATE?
612
YES
RETURN
614
NO
HANDLE ALL OTHER REQUESTS
616

Fig. 6

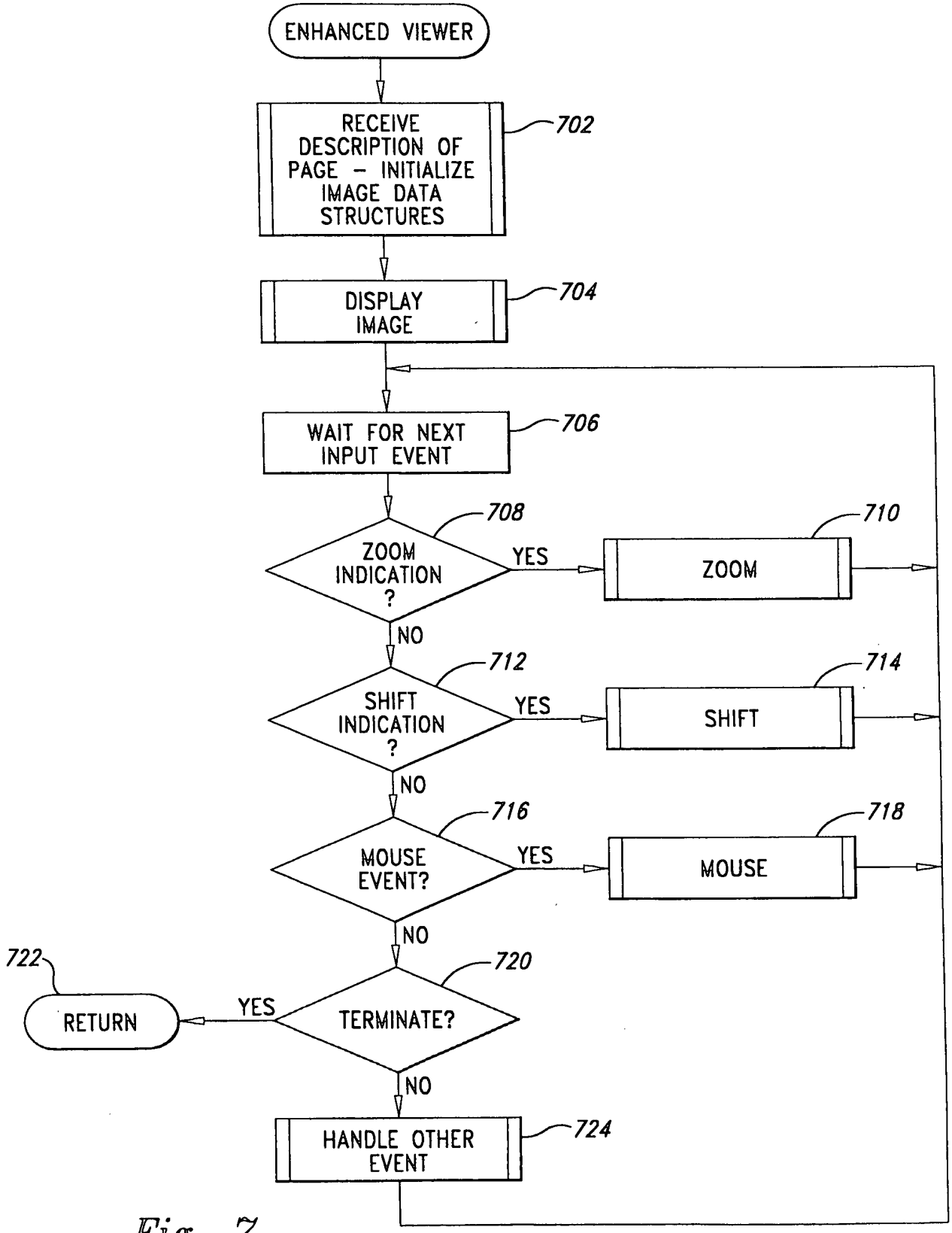

ENHANCED VIEWER
RECEIVE DESCRIPTION OF PAGE – INITIALIZE IMAGE DATA STRUCTURES
702
DISPLAY IMAGE
704
WAIT FOR NEXT INPUT EVENT
706
ZOOM INDICATION ?
708
YES
ZOOM
710
NO
SHIFT INDICATION ?
712
YES
SHIFT
714
NO
MOUSE EVENT?
716
YES
MOUSE
718
NO
TERMINATE?
720
YES
RETURN
722
NO
HANDLE OTHER EVENT
724

Fig. 7

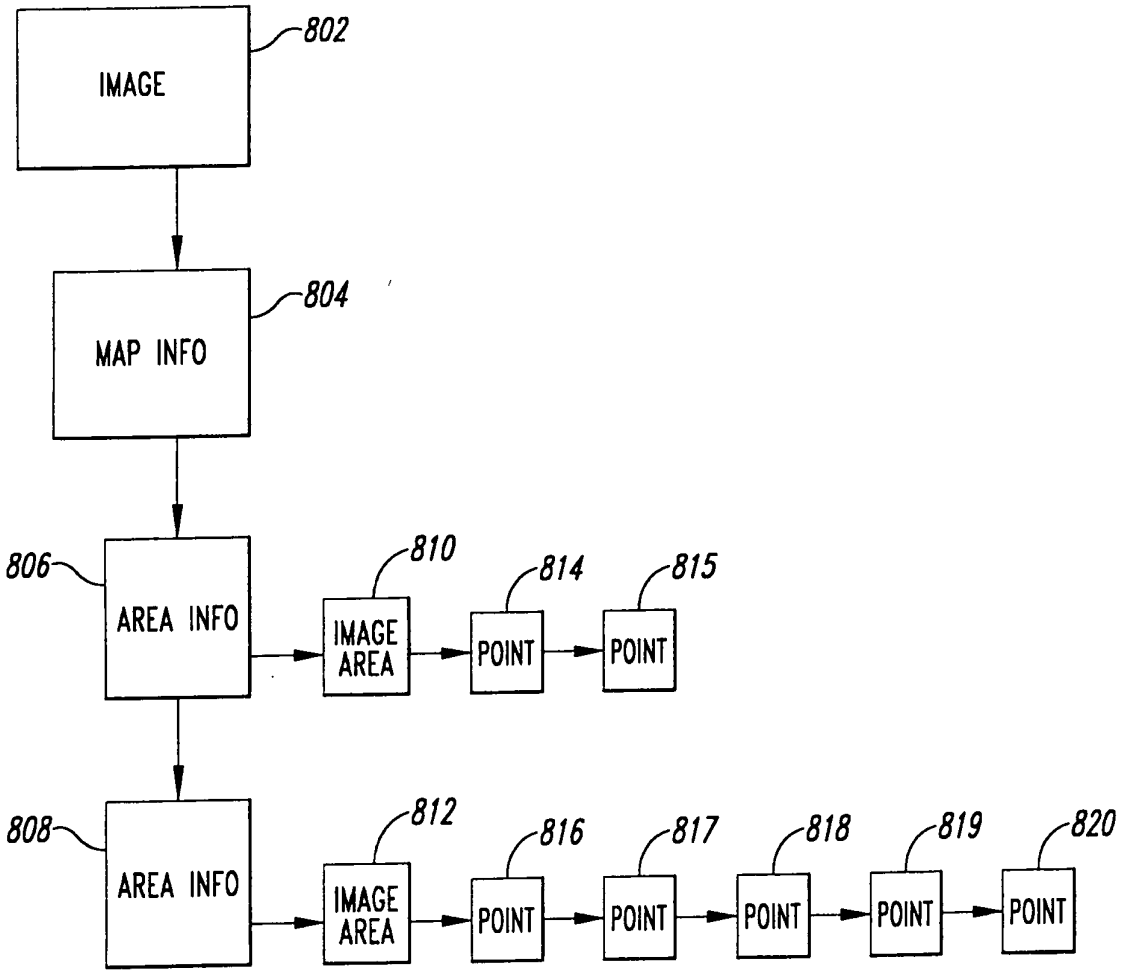


IMAGE
802
MAP INFO
804
806
AREA INFO
810
IMAGE AREA
814
POINT
815
POINT
808
AREA INFO
812
IMAGE AREA
816
POINT
817
POINT
818
POINT
819
POINT
820
POINT

Fig. 8

APPENDIX III

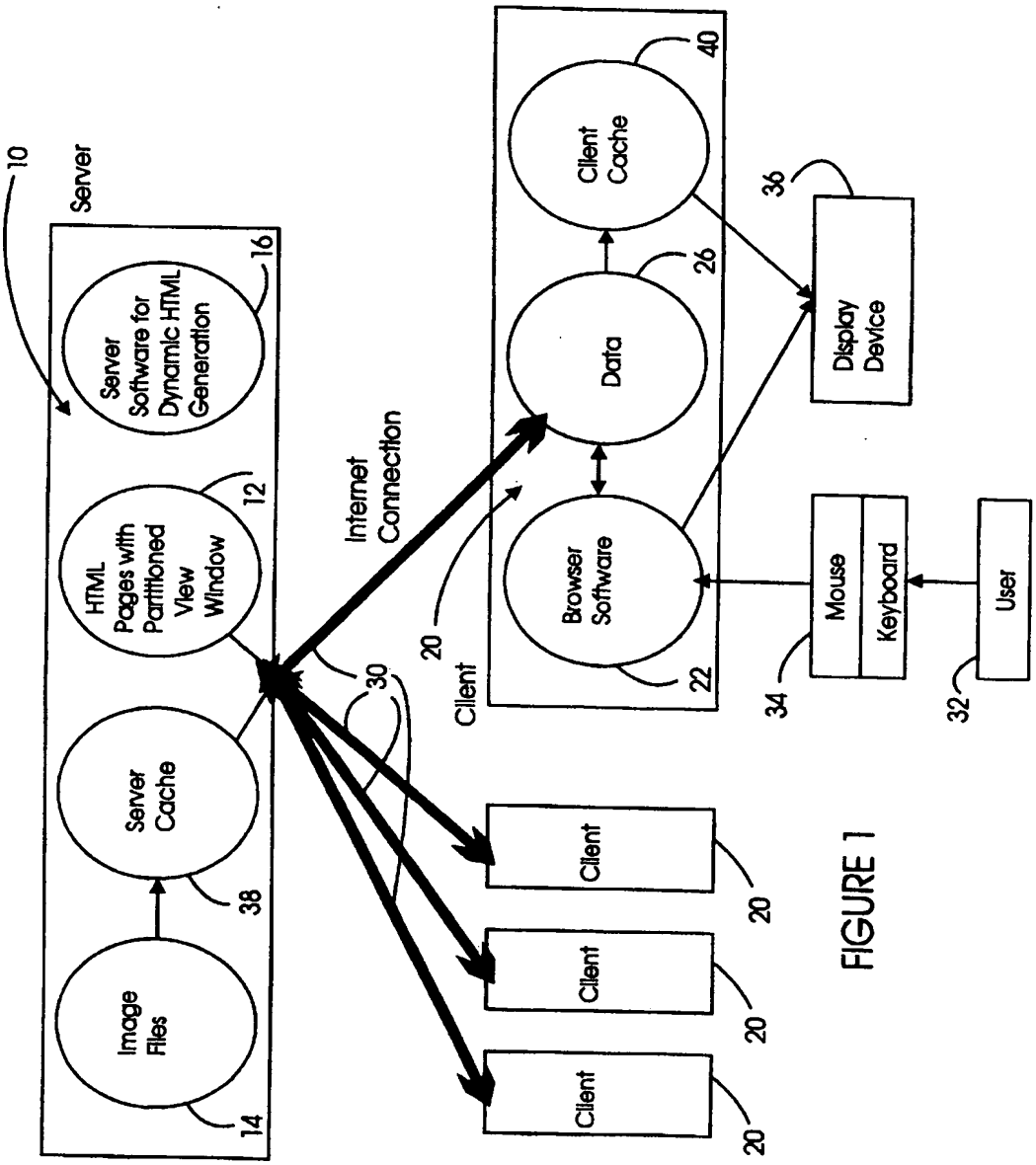

10
Server
Server Software for Dynamic HTML Generation
16
HTML Pages with Partitioned View Window
12
Server Cache
38
Image Files
14
Internet Connection
30
Client
20
Client Cache
40
Data
26
Browser Software
22
Display Device
36
Mouse
Keyboard
34
User
32
Client
20
Client
20
Client
20

FIGURE 1

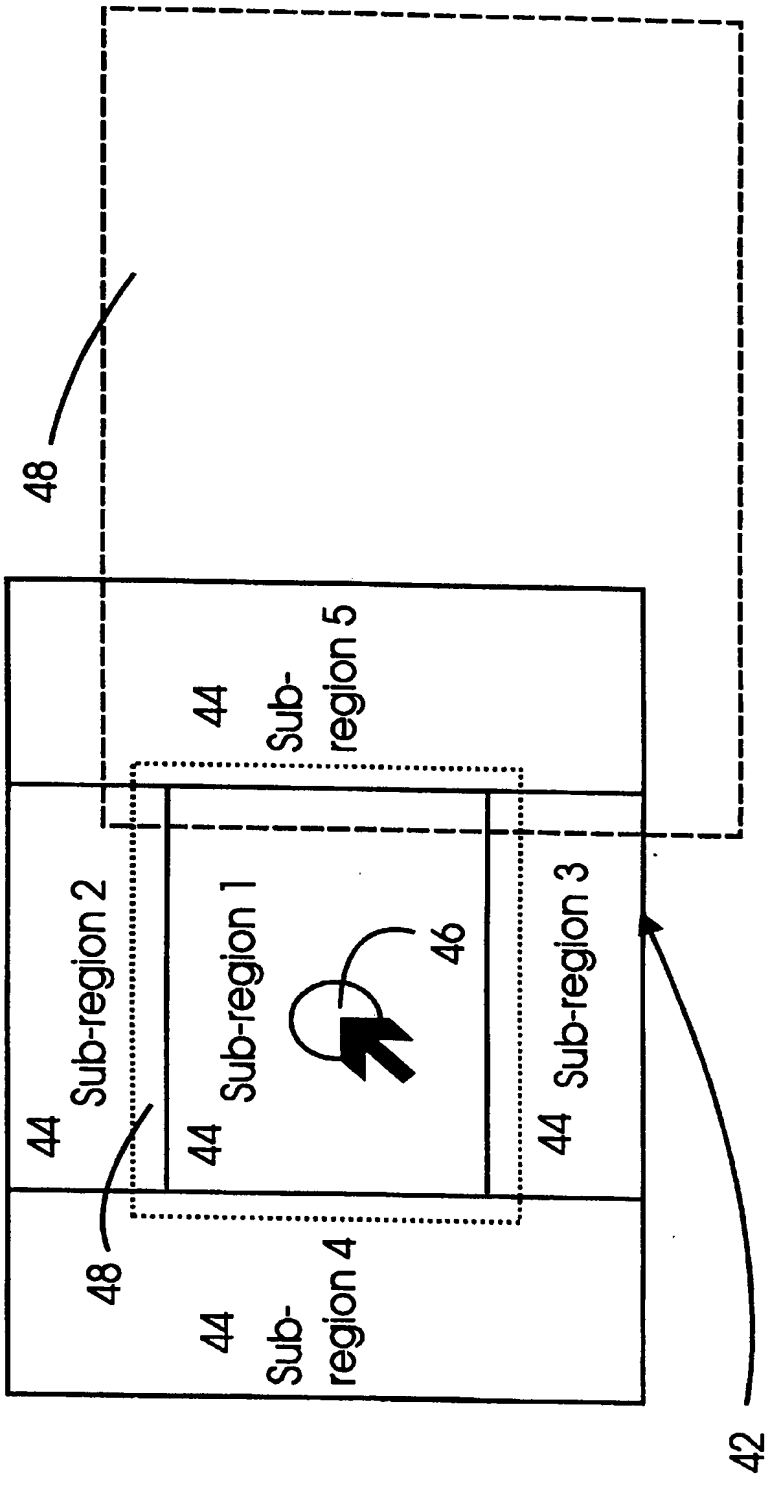

48
44
Sub-region 5
Sub-region 2
Sub-region 1
46
Sub-region 3
44
44
44
48
44
Sub-region 4
42

FIGURE 2

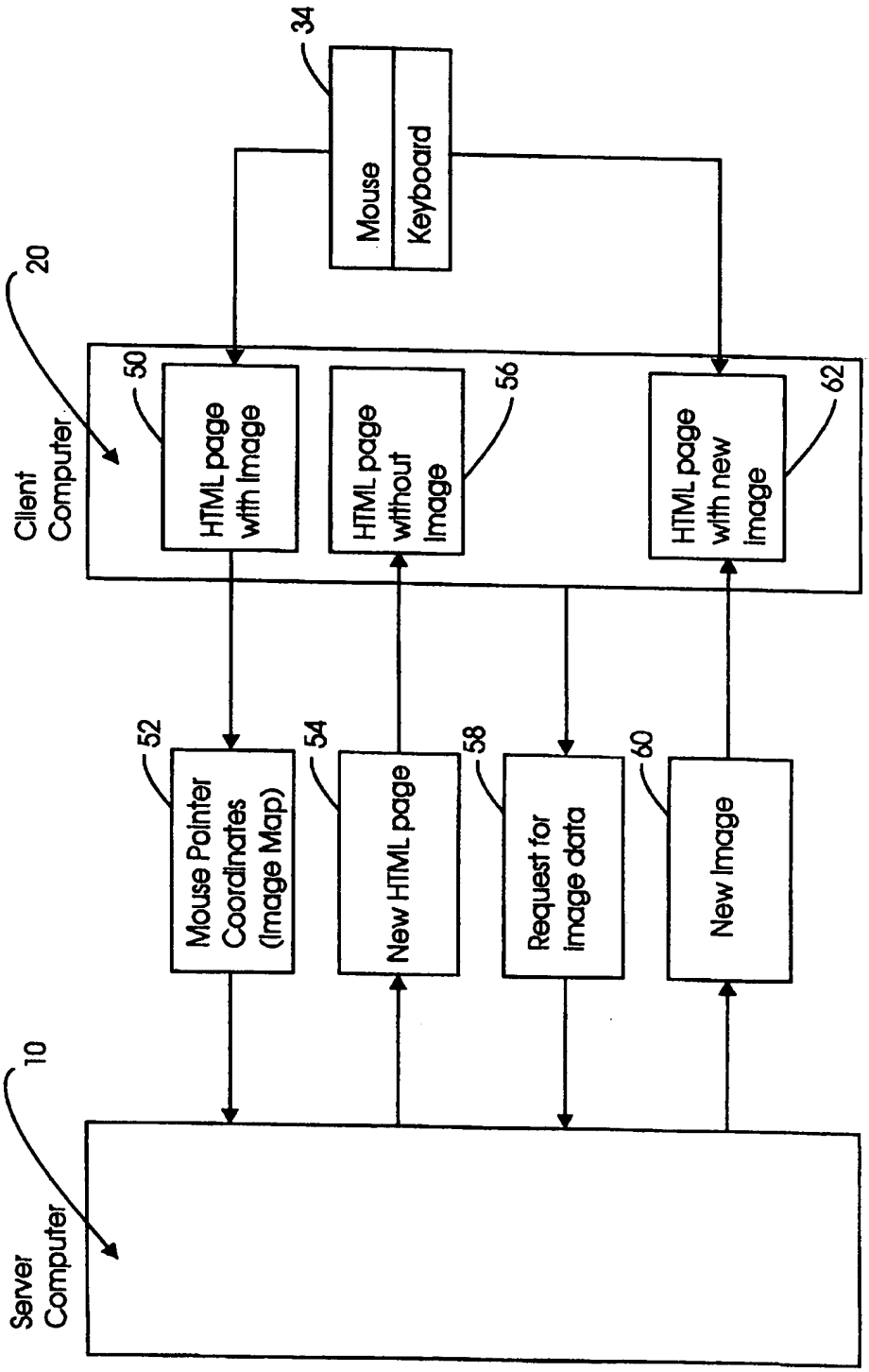

34
Mouse
Keyboard
20
Client Computer
50
HTML page with image
HTML page without image
56
HTML page with new image
62
52
Mouse Pointer Coordinates (Image Map)
54
New HTML page
58
Request for image data
60
New Image
10
Server Computer

FIGURE 3

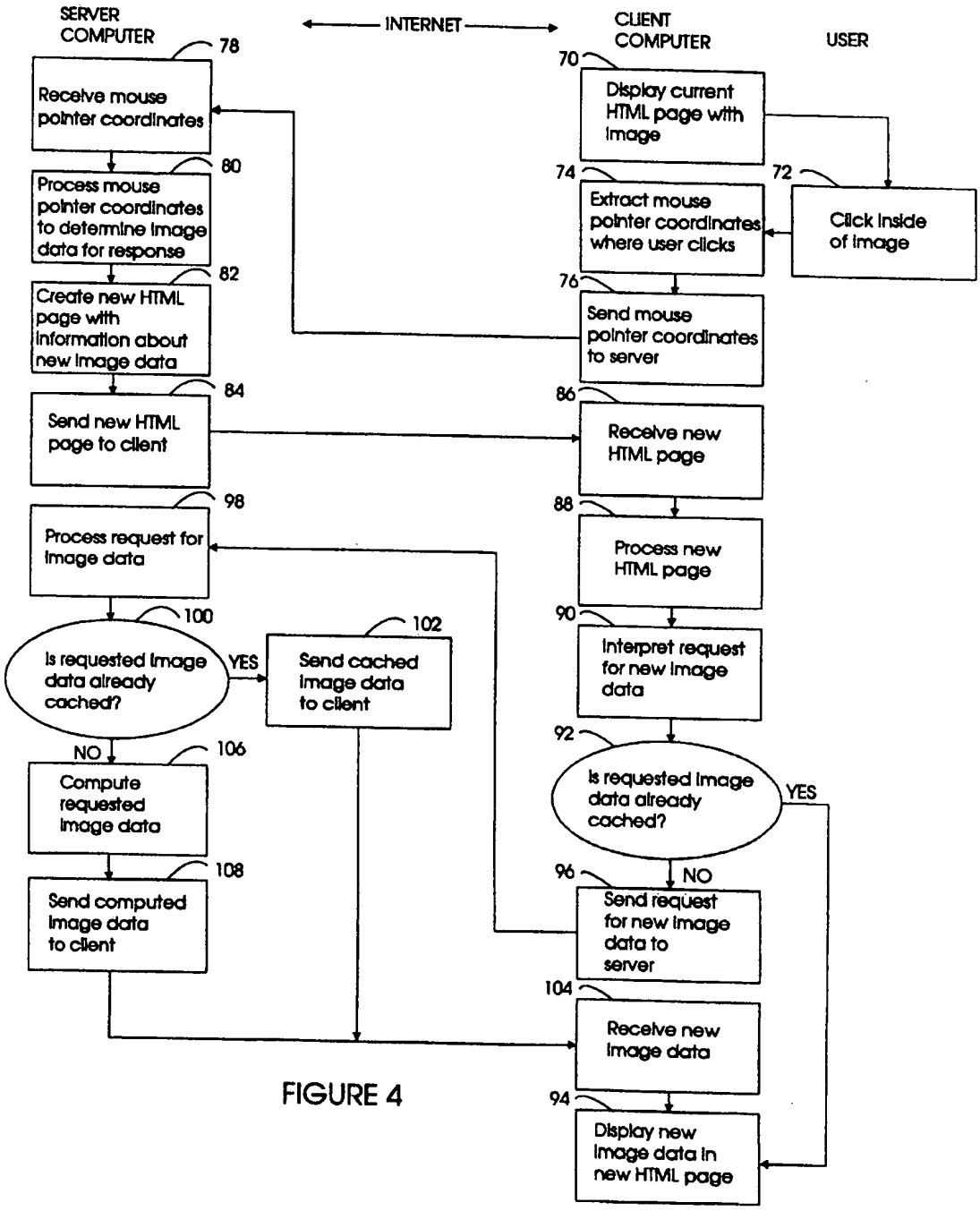

SERVER COMPUTER
INTERNET
CLIENT COMPUTER
USER
78
Receive mouse pointer coordinates
80
Process mouse pointer coordinates to determine image data for response
82
Create new HTML page with information about new image data
84
Send new HTML page to client
98
Process request for image data
100
Is requested image data already cached?
YES
102
Send cached image data to client
NO
106
Compute requested image data
108
Send computed image data to client
70
Display current HTML page with image
74
Extract mouse pointer coordinates where user clicks
72
Click inside of image
76
Send mouse pointer coordinates to server
86
Receive new HTML page
88
Process new HTML page
90
Interpret request for new image data
92
Is requested image data already cached?
YES
NO
96
Send request for new image data to server
104
Receive new image data
94
Display new image data in new HTML page

FIGURE 4

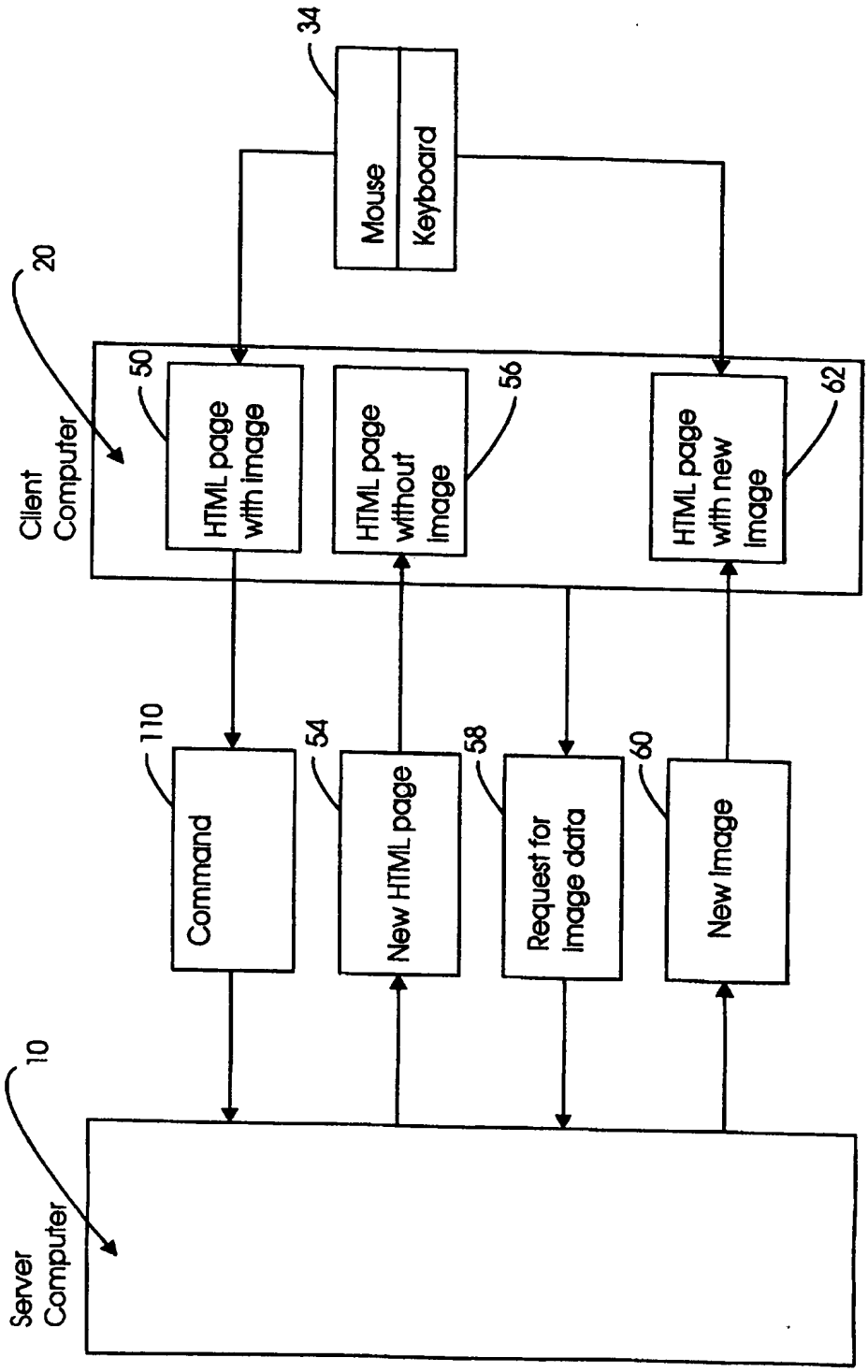

34
Mouse
Keyboard
20
Client Computer
50
HTML page with image
HTML page without image
56
HTML page with new image
62
110
Command
54
New HTML page
58
Request for image data
60
New Image
10
Server Computer

FIGURE 5

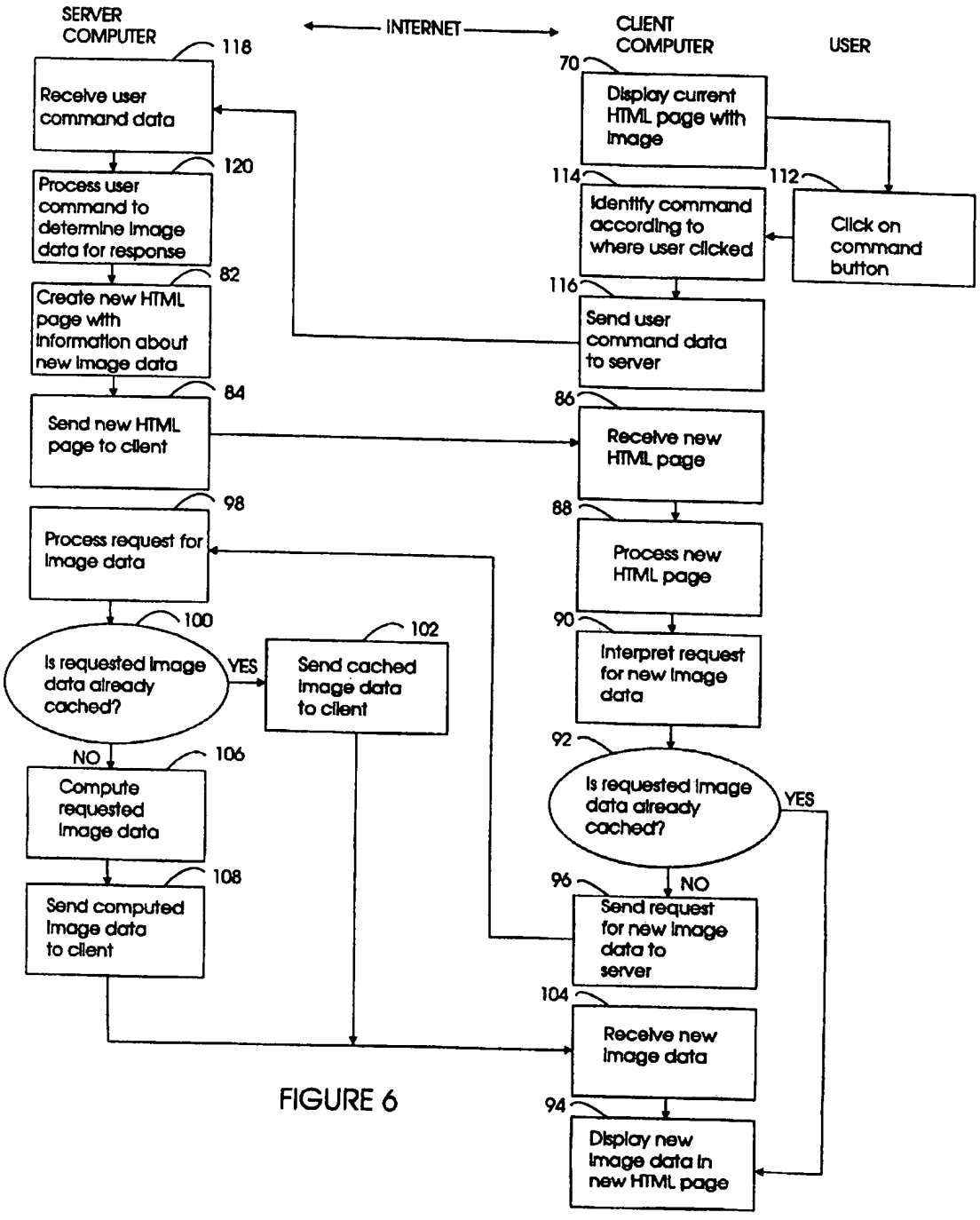

SERVER COMPUTER
INTERNET
CLIENT COMPUTER
USER
118
Receive user command data
120
Process user command to determine image data for response
82
Create new HTML page with information about new image data
84
Send new HTML page to client
98
Process request for image data
100
Is requested image data already cached?
YES
102
Send cached image data to client
NO
106
Compute requested image data
108
Send computed image data to client
70
Display current HTML page with image
114
Identify command according to where user clicked
112
Click on command button
116
Send user command data to server
86
Receive new HTML page
88
Process new HTML page
90
Interpret request for new image data
92
Is requested image data already cached?
YES
NO
96
Send request for new image data to server
104
Receive new image data
94
Display new image data in new HTML page

FIGURE 6

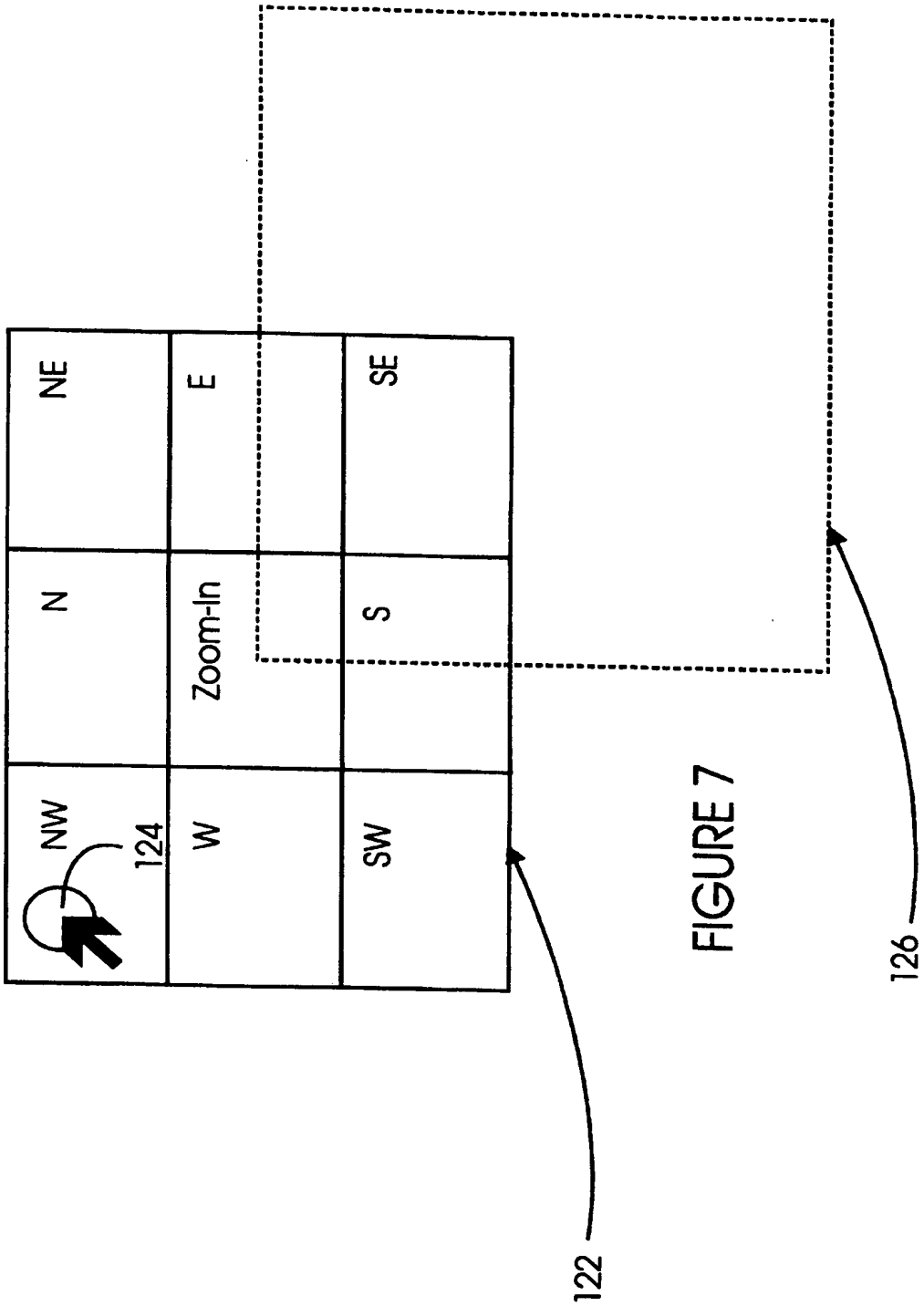


FIGURE 7

PATENT

I hereby certify that on the date specified below, this correspondence is being deposited with the United States Postal Service as first-class mail in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Jan. 20, 2004 — Joanne Bourguignon

Date — Joanne Bourguignon

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants:	Lawrence Bain et al.
Application No.:	09/298,453
Filed:	April 13, 1999
Title:	DYNAMIC-ADAPTIVE CLIENT-SIDE IMAGE MAP
Examiner:	Cong Lac Huynh
Art Unit:	2178
Docket No.:	10990633-1
Date:	January 16, 2004

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT AFTER FINAL

INTRODUCTORY COMMENTS

Sir:

In response to the Final Office Action dated July 15, 2003, please amend the application as follows:

AMENDMENTS TO THE CLAIMS

1. (original) A method for associating an active region with a corresponding position within an image included in a page displayed by a browser running on a client computer, the method comprising:

sending a request by the browser to a server for a description of a page that includes a specification of the image and an associated client-side image map, the client-side image map specifying a shape, size, and location of the active region within the image and specifying actions to be performed in response to input events directed to the active region;

receiving from the server in response to the request a description of the requested page that includes an invocation of a viewer for displaying the image, the invocation including parameters that describe the image and the client-side image map;

instantiating the viewer and passing to the viewer the parameters included in the invocation;

storing by the viewer representations of active regions within the image in image-relative coordinates along with indications of the actions to be performed in response to input events directed to the active region; and

when an input event is detected by the browser during display of the page,

passing the input event by the browser to the viewer, and

when the viewer determines that the input event was input to a position within the image corresponding to the active region, determining an action specified for performance in response to the input event to the active region and calling for performance of the determined action.

2. (original) The method of claim 1 wherein the page displayed by the browser running on a client computer is a web page.

3. (original) The method of claim 2 wherein the server runs on a server computer and a description of the web page is requested by the browser from the server and received by the browser from the server via the Internet.

4. (original) The method of claim 2 wherein the server runs on the client computer and a description of the web page is requested by the browser from the server and received by the browser from the server via an inter-process communications medium within the client computer.

5. (original) The method of claim 2 wherein the description of the web page received from the server in response to the request by the browser is a hyper-text markup language document.

6. (original) The method of claim 2 wherein the image is an OpenPix image and wherein an invocation to a browser extension image viewer is included in the description of the web page.

7. (original) The method of claim 2 wherein input events directed to the active region may include mouse-click, mouse-into, and mouse-out-from events, and actions to be performed in response to input events include display of a web page, display of an image, or launching of a software routine.

8. (previously amended) The method of claim 2 where image-relative coordinates represent the position of points within the image, a point within the image represented by a pair of coordinates, a first coordinate of the pair having a fractional value representing the ratio of a horizontal line segment to a horizontal dimension of the image with a first endpoint coincident with a vertical edge of the image and a second endpoint coincident with the point, the horizontal line segment perpendicular to the vertical edge of the image, the second coordinate of the pair having a fractional value representing the ratio of a vertical line segment to a vertical dimension of the image with a first endpoint coincident with a horizontal edge of the image and a second endpoint coincident with the point, the vertical line segment perpendicular to the horizontal edge of the image, the horizontal and vertical edges of the image intersecting at an origin having coordinates (0, 0).

9. (original) The method of claim 2 further including:

when a display altering input event is detected by the browser,

passing a display altering input command by the browser to the viewer, and

altering the display of the image by the viewer in accordance with the input command.

10. (original) The method of claim 9 wherein display altering input events include a zoom input event and a pan input event.

11. (previously amended) A method for serving a description of a page from a server to a browser running on a client computer that requests the page, the description of the page provided to the browser by the server containing an invocation of a viewer, the invocation including parameters that specify an image included in the page and an active region within the image, the method comprising:

receiving a request from the browser by the server for a description of the page that includes a specification of the image and an associated client-side image map, the client-side image map specifying a shape, size, and location of the active region within the image and that specifies actions to be performed in response to input events directed to the active region;

retrieving a description of the page;

determining the capabilities for viewing pages provided by the browser running on the client computer; and

when the browser, running on the client computer, is capable of accepting display altering commands from a user while displaying a page,

parsing the description of the page to find the specification of the image and the client-side image map included in the page,

substituting, in the description of the page, an invocation of a viewer for the specification of the image and the client-side image map included in the page, including in the invocation parameters that specify the image and the client-side image map, to create a transformed page description, and

sending the transformed page description to the browser.

12. (original) The method of claim 11 wherein the page requested from the server by the browser running on a client computer is a web page.

13. (original) The method of claim 12 wherein the server runs on a server computer and a description of the web page is requested by the browser from the server and sent by the server to the browser via the Internet.

14. (original) The method of claim 12 wherein the server runs on the client computer and a description of the web page is requested by the browser from the server and sent by the server to the browser via an inter-process communications medium within the client computer.

15. (original) The method of claim 12 wherein the description of the web page retrieved by the server in response to the request by the browser is a hyper-text markup language document.

16. (original) The method of claim 12 wherein the image is an OpenPix image and wherein an invocation to a browser extension image viewer is included in the description of the web page.

17. (original) The method of claim 12 wherein input events directed to the active region may include mouse-click, mouse-into, and mouse-out-from events, and actions to be performed in response to input events include display of a web page, display of an image, or launching of a software routine.

18. (currently amended) A system for displaying a page that includes an image and an active region correlated with a particular portion of the image, the display of the page modifiable during the display of the page on a display device of a client computer such that the active region within the image remains correlated with the portion of the image, the system comprising:

a browser running on the client computer that displays the page;

a server that receives a request from the browser for a description of the page and that that provides a description of the page that contains an invocation of a viewer, the invocation including parameters that specify an image included in the page and an active region within the image; and

data structures on the client computer that store image-relative indications of the particular portion of the image associated with the active region and actions ~~and actions~~ to be performed in response to input events directed to the active region.

19. (currently amended) The ~~method~~ system of claim 19 wherein the page displayed by the browser running on a client computer is a web page.

20. (currently amended) The ~~method~~ system of claim 19 wherein the server runs on a server computer and a description of the web page is requested by the browser from the server and received by the browser from the server via the Internet.

21. (currently amended) The ~~method~~ system of claim 19 wherein the server runs on the client computer and a description of the web page is requested by the browser from the server and received by the browser from the server via an inter-process communications medium within the client computer.

22. (currently amended) The ~~method~~ system of claim 19 wherein the description of the web page received from the server in response to the request by the browser is a hyper-text markup language document.

23. (currently amended) The ~~method~~ system of claim 19 wherein input events directed to the active region may include mouse-click, mouse-into, and mouse-out-from events, and actions to be performed in response to input events include display of a web page, display of an image, or launching of a software routine.

24. (currently amended) The ~~method~~ system of claim 19 where image-relative coordinates represent the position of points within the image, a point within the image represented by a pair of coordinates, a first coordinate of the pair having a fractional value representing the ratio of a horizontal line segment to a horizontal dimension of the image with a first endpoint coincident with a vertical edge of the image and a second endpoint coincident with the point, the horizontal line segment perpendicular to the vertical edge of the image, the second coordinate of the pair having a fractional value representing the ratio of a vertical line segment to a vertical dimension of the image with a first endpoint coincident with a

horizontal edge of the image and a second endpoint coincident with the point, the vertical line segment perpendicular to the horizontal edge of the image, the horizontal and vertical edges of the image intersecting at an origin having coordinates (0, 0).

REMARKS

Please consider the above amendments made in order to correct a number of typographical errors and to place the claims into proper form for the Appeal.

Respectfully submitted,
Lawrence Bain et al.
Olympic Patent Works PLLC

Robert W. Bergstrom
Registration No. 39,906

Olympic Patent Works PLLC
P.O. Box 4277
Seattle, WA 98194-0277
206.621.1933 telephone
206.621.5302 fax